
150 ptas.
mi computer 5
CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR
GHOSTS AND WITCHES
Jamie
081 Software educativo
084 Código de los ordenadores
086 Historia de la informática
089 BBC Modelo B
092 Puertas y sumadores
094 Cassettes
096 RAM y ROM
098 Programación Basic
Editorial Delta, S.A.
SONY

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen I - Fascículo 5

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Roberto Quiroga

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.° - Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-84-6 (tomo 1)
84-85822-82-X (obra completa)
Depósito Legal: B. 52-84

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 158402
Impreso en España - Printed in Spain - Febrero 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

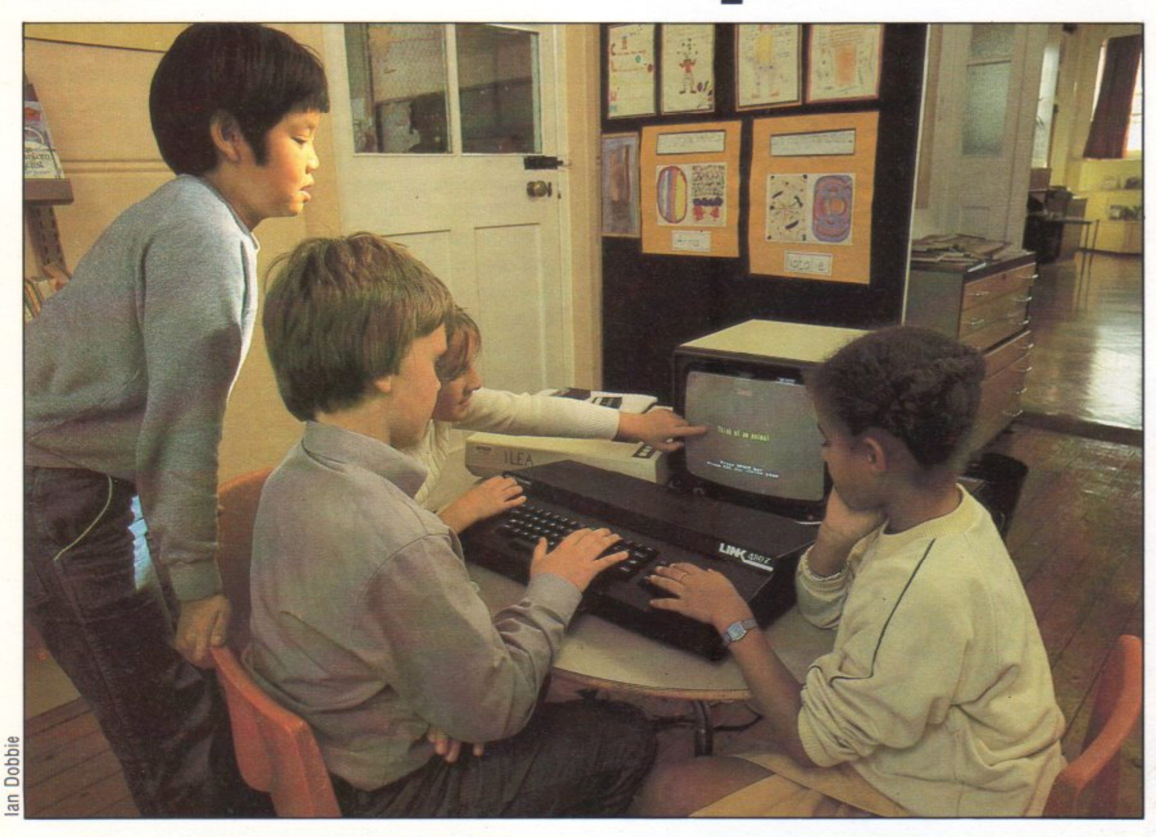
Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual, sin variación alguna sobre el precio de venta en vigor en el momento de la petición. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



Escuela en la pantalla



Ian Dobbie

El ordenador en la enseñanza básica
Diversos países se están preparando para el futuro mediante la aplicación de una política educativa que tiende a familiarizar a los niños con los ordenadores ya durante sus primeros años de escolarización. No se trata tan sólo de familiarizar al niño con la nueva tecnología, sino también de utilizar el ordenador para enseñar una gran variedad de temas, desde biología hasta lenguas extranjeras. Un ordenador es un maestro paciente e ideal, ya que no introduce un nuevo concepto hasta que el niño no domine por completo el problema anterior, permitiendo que tanto el alumno de comprensión lenta como el más despierto aprendan al ritmo de sus propias posibilidades. Como medio auxiliar de aprendizaje, el ordenador ofrece una gama de aplicaciones más amplias: por el solo hecho de utilizarlo, el niño descubre cómo se analiza y se resuelve un problema

La era de los ordenadores ya ha comenzado en las aulas, y existe en el mercado una fascinante y variada gama de programas educativos para niños de todas las edades

A finales de 1983 había en España más de 4 000 ordenadores personales dedicados a la enseñanza, y se estima que dentro de pocos años la mayoría de los centros de educación primaria y secundaria, así como también los de formación profesional, contarán al menos con un microordenador. En la actualidad los ordenadores se utilizan no sólo para impartir "nociones de informática", sino también, en el campo educativo, para enseñar temas relacionados con las matemáticas y la alfabetización, ayudar a los niños de comprensión lenta y estudiar lenguas extranjeras.

En el mercado existen muchos programas educativos para ordenadores personales, pero los maestros suelen quejarse de su poca calidad. Ello se debe a que son muy pocos los programas que se han escrito respetando por igual tanto la disciplina educativa como la informática.

Es muy raro que un programador de ordenadores posea experiencia en el campo de la enseñanza; y los maestros, muchos de los cuales apenas iniciados en este campo, incurren a veces en los más garrafales errores de programación.

Aunque lo más probable es que el programa de un maestro sólo se aplique en su propia clase, los problemas surgen en cuanto dicho programa se envía a otra escuela. Por lo general, el programa en sí mismo, almacenado ya sea en cassette o en disco, no es suficiente, haciéndose imprescindible una buena documentación explicativa, pues sin ella es posible que los estudiantes sean incapaces de hacer funcionar el programa. Ante este problema, tal vez la respuesta del programador fuera: "¡Por supuesto, *se entiende* que se ha de digitar LOAD (cargar)!", pero a quien no sabe nada acerca de los ordenadores se le han de explicar uno por uno todos estos detalles.

Pasando a un plano más concreto, una buena programación requiere anticiparse a todos los errores que pueda cometer un principiante. Esto es muy importante para asegurar que el programa sea, efectivamente, un buen medio auxiliar de enseñanza. Una buena programación consiste en algo más que eliminar de un programa todos los posibles márgenes de error, hasta el punto de que sólo ejecute lo que debería hacer exclusivamente cuando se pulse la tecla adecuada. La buena programación también debe asegurar que el programa no hará nada que no deba realizar cuando se pulse una tecla equivocada. Ésta es la parte más complicada de la escritura de un programa. Éste ha de ser capaz de recuperarse de los más flagrantes errores en que pueda incurrir un niño y, al mismo tiempo, inculcarle la idea de que usar un ordenador es algo sencillo y divertido.

A pesar de estos problemas, existe una amplia gama de programas educativos tanto para la escuela como

para el hogar. Un ordenador es una maravillosa herramienta educativa y, para acertar en la elección del software más adecuado para sus hijos, le será de gran utilidad conocer las diversas maneras en que se puede emplear.

Un ordenador puede utilizarse para instruir al niño en prácticamente cualquier tema. Si el programa es bueno, lo más probable es que el niño se sienta fascinado y muy motivado para aprender.

La clase de programas educativos más generalizada se conoce como "de procedimiento correcto y ejercicios". En un programa de este tipo, al niño se le proporcionan ejemplos y luego se le invita a resolver problemas similares. Normalmente el programa lleva la cuenta de los fallos y los aciertos del niño. Incluso estimula al alumno cuando éste da una respuesta correcta y le sugiere amablemente "inténtalo otra vez" cuando la respuesta es equivocada.

Para decidir qué programas son los más adecuados para su hijo, deberá tener en consideración diversos factores: la edad del niño, el modelo de ordenador que usted haya adquirido y lo que su hijo esté estudiando en la escuela.

Si aún no ha comprado un ordenador personal, pero piensa que cuando lo haga una de las funciones a que lo destinará en su hogar será la educación, valdrá la pena averiguar qué tipo de ordenador usa su hijo en la escuela. Si usted está en condiciones de comprar un modelo similar, su hijo podrá realizar en casa los mismos programas educativos que utiliza en el colegio. A muchas escuelas les interesa proporcionar a los padres copias de los programas que se emplean en clase, y estos "deberes" pueden representar una considerable ayuda. Si usted ya ha adquirido un ordenador personal y éste no es compatible con los ordenadores de la escuela, no se preocupe; la experiencia que adquiera su hijo al digitar ordenadores diferentes también será muy valiosa.

Es natural que la mayoría de los programas educativos existentes hayan sido creados para las marcas de ordenadores más competitivas del mercado, pero muchos fabricantes están potenciando el papel docente del ordenador. Existe una gama particularmente amplia de programas educativos para el Apple, el Commodore PET, el Tandy, el BBC Micro, Sinclair y Texas Instruments; no obstante, algunos de los fabricantes de software incorporados recientemente a este mercado aún deben aportar una oferta de programas realmente más variada.

Los programas educativos para cualquier ordenador pueden conseguirse a través de la empresa fabricante o por medio de diversas firmas productoras de software independientes. Estas últimas escriben programas para ordenadores y anuncian sus productos en las revistas de informática y en los establecimientos que venden ordenadores personales.

La elección adecuada

Los programas destinados a niños menores de ocho años se centran en la enseñanza de las operaciones aritméticas básicas y en la alfabetización. Una de las gamas más interesantes de programas educativos para niños pequeños es la que produce Texas Instruments. El ordenador personal TI-99/4A, por ejemplo, fue muy bien acogido por los padres norteamericanos y británicos, pues contaba con una amplia gama de programas educativos de TI producida tanto por Texas Instruments como por Scott, Foresman & Co., de Estados Unidos. El 99/4A es un ordenador de 16 bits, lo que significa que probablemente los programas escritos en código de lenguaje máquina sean mucho mejores que los programas escritos para la mayoría de los ordenadores personales corrientes de 8 bits.

Esto lo demuestran algunos programas de TI tales como *Beginning grammar* (Iniciación a la gramática), *Addition and subtraction* (Suma y resta) y *Number magic* (La magia de los números). Estos programas van almacenados en un cartucho plástico que se introduce en el TI-99/4A, por lo cual a los niños les resultan muy manejables. No obstante, el hecho de que el software sea norteamericano representa un problema, y



El mejor compañero
El microordenador ayuda a resolver los "deberes", acompaña en los ratos de ocio y proporciona amigos. Pero para obtener un buen rendimiento de esta maravillosa máquina, es necesario acertar en la elección de los programas adecuados de acuerdo con la edad de los niños

Cortesía de Investrónica, S.A.

algunos de los programas suponen un gran inconveniente para los maestros de otros países. Sin embargo, TI posee un magnífico ejemplo de LOGO (véase p. 34), si bien realmente se encuadra en la segunda categoría de herramientas de descubrimiento.

Para niños de más edad se cuenta con un software muy rico y variado. Los programas para edades entre 8 y 11 años se caracterizan por su complejidad y calidad, y la mayoría de ellos tienen como objetivo reforzar los conocimientos básicos del alumno y poner a prueba sus aptitudes. A esta edad el niño comienza a interesarse por otros temas, como la música y las lenguas extranjeras, que pueden enseñarse, asimismo, por ordenador. Existen programas de esta naturaleza para la mayoría de los aparatos.

Para alumnos de enseñanza media existe una multitud de programas. Para hacer una selección de ellos y escoger sólo los mejores, lo más indicado es que hable con el profesor de su hijo.

Es muy importante que el trabajo escolar que realice el niño en su casa siga la misma línea que el que lleva a cabo en la escuela; la mayoría de los profesores se mostrarán bien dispuestos para orientar y ayudar a los padres en la elección del tipo de software educativo más apropiado para sus hijos.

Existe aún otra categoría de programas educativos para ordenador, destinada fundamentalmente a los niños menores de 13 años. A esta edad los niños aún están descubriendo la forma de aprender y para ello resultan muy valiosos los programas que los introducen en el uso del ordenador como medio de descubrir el mundo por sí mismos. El programa más conocido de esta categoría es el LOGO, lenguaje del cual existen versiones para ordenadores de las firmas Atari, Tandy, Apple, Texas Instruments, Research Machines, Commodore e IBM. Por su parte, Sinclair y BBC han mostrado interés, también, por realizar sus versiones, pero aún no se encuentran en el mercado. A través de este programa, el niño entre 6 y 12 años de edad es estimulado para explorar el potencial gráfico del ordenador (y, a su vez, la geometría), mediante una "tortuga". El niño descubre cómo enseñarle a la tortuga a recordar procedimientos (programas) y, en determinados casos, puede ir progresando hasta llegar a dibujar en la pantalla todo un mundo de fantasía. Al utilizar estos programas, lo que el niño hace realmente es enseñarse a sí mismo las leyes básicas de las matemáticas y, en este sentido, se ha reconocido claramente la efectividad de este programa para ayudar a comprender conceptos matemáticos y espaciales.

Escoger un buen programa educativo no es una labor fácil, debido a la gran cantidad de software disponible. Es muy interesante asistir a alguna de las ferias de informática que se celebran periódicamente en el país. Con toda probabilidad, allí se encontrará con los propios fabricantes y programadores exhibiendo sus productos y, de este modo, tendrá la ocasión de conocer, probar y comparar programas.

Existe la opinión generalizada de que la actual escasez de buenos programas, que satisfagan tanto las exigencias relativas al ordenador como a la educación en sí misma, no ha de durar mucho tiempo más. Todos los meses aparecen programas nuevos que, con toda seguridad, serán un valioso estímulo para el desarrollo intelectual de sus hijos. Conviene tener presente a este respecto que, en un próximo futuro, la informática será indispensable en todos los campos de la actividad humana.

El Agente Secreto

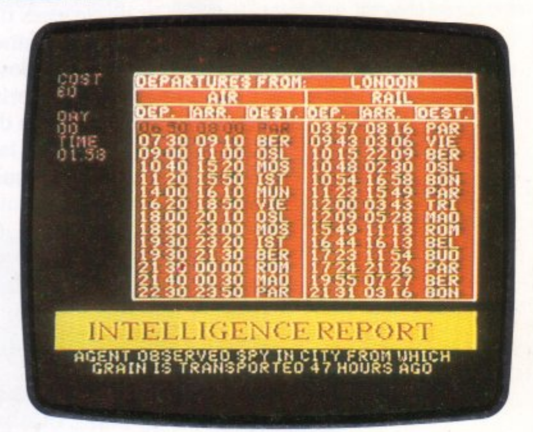
USTED ES UN AGENTE
DE MI6

SU MISIÓN ES CAPTURAR
A UN FAMOSO ESPÍA

OBJETIVO ESPÍA: ELIMINAR
SUS AGENTES EN
CADA CIUDAD

Actualmente existen muchos programas educativos en el mercado. El que muestran las fotografías se denomina *Agente Secreto* y ha sido realizado por Heinemann. Su misión consiste en capturar a un famoso espía enemigo que se halla en Europa, antes de que él elimine a todos sus agentes. Para poder atraparlo, se han de descifrar las claves acerca de su paradero. Comienza la búsqueda...

Al espía sólo se le puede capturar en una ciudad; pero él no permanece en el mismo sitio durante más de dos horas. Cuando cree saber dónde se halla éste, usted tiene la alternativa de dirigirse hasta allí en tren o en avión. Deberá decidir qué es más importante: llegar rápido o el precio a pagar





Cada vez que alguno de los agentes que trabajan para usted le envíe un mensaje, se encenderá una luz intermitente en el mapa que le indicará desde qué ciudad proviene. Si su agente fuera eliminado, el mensaje será interceptado antes de que usted lo reciba. En ese caso, tal vez le interese contratar los servicios de un nuevo agente; pero entonces habrá de pagar por él...

Los informadores estarán felices de venderle información, pero ésta se envía en clave, por lo cual quizá le interese entregársela a un especialista para que le ayude a descifrarla. Al final del juego habrá aprendido los nombres y la situación de las más importantes ciudades europeas, entenderá los anuncios de salidas y llegadas en estaciones y aeropuertos y será un experto en la confección de presupuestos

TARDÓ 27 DÍAS 13 HORAS
EN CAPTURAR ESPÍA

GASTÓ 1 500 000 PTAS.

QUEDA DESPEDIDO DE M16

Ian McKinnell

Descifrando el código

Digite su lenguaje de ordenador en el teclado y un programa que funciona dentro del micro lo convertirá rápidamente en su propio código de lenguaje máquina

A pesar de que parece que todos los microordenadores desempeñan funciones similares, cada modelo es exclusivo. Algunos se venden con programas ya incorporados, mientras que otros requieren que estos programas se les "lean" desde un disco o una cinta de cassette exteriores.

Algunas máquinas poseen un único programa global que permite tanto la entrada de programas como la utilización de órdenes directas como SAVE o LOAD. Para realizar estas dos funciones, otros modelos necesitan programas separados.

No obstante, la mayoría de los microordenadores más conocidos funcionan de acuerdo a principios similares. El movimiento de la información hacia y desde dispositivos de almacenamiento externos (discos y cintas) a la pantalla se controla, en ambos casos, mediante el teclado. Asimismo, todas las máquinas pueden comunicarse con otros dispositivos externos, como impresoras, trazadores de gráficos e instrumentos científicos. Y la mayoría de los micros admiten que sus usuarios escriban programas en lenguajes similares al corriente, como el BASIC.

Cuando digita en el teclado de un ordenador un programa en BASIC, otro programa denominado *sistema operativo* pasa lo que ha digitado tanto a la pantalla como a un programa interpretador de BASIC. Esto significa que en el interior del ordenador se están llevando tres programas simultáneamente: el sistema operativo, el interpretador de BASIC y su propio programa.

Cuando usted ejecuta su programa, estos tres programas deben actuar al mismo tiempo. Cada instrucción en BASIC de su programa es traducida por el interpretador, y las instrucciones resultantes en código de lenguaje máquina pasan, una por una, al microprocesador para que las ejecute. Al mismo tiempo, el sistema operativo está explorando el teclado para detectar la entrada de datos y poder visualizarlos en pantalla.

Si una de las instrucciones de su programa solicita que algo se imprima o se grabe en disco, por ejemplo, el interpretador le requeriría al sistema operativo que realice esta tarea.

La ilusión de que en el mismo momento se están produciendo diversas acciones se crea por la increíble velocidad con que trabaja el microprocesador. Éste procesa las instrucciones del sistema operativo y del interpretador con tanta rapidez que ambas pueden ejecutarse simultáneamente.

Algunas máquinas trabajan a velocidad aún mayor, permitiendo la llegada de datos para "interrumpir" el procesamiento normal. De esta forma, no es necesario que el sistema operativo verifique las diversas fuentes de entradas de datos externos, como el teclado o las unidades de disco.

Se denomina *editor* a un tipo de procesador de textos menos sofisticado. La calidad de los editores suele variar considerablemente; no obstante, es probable que el editor incorporado del interpretador de BASIC sea de similar calidad.

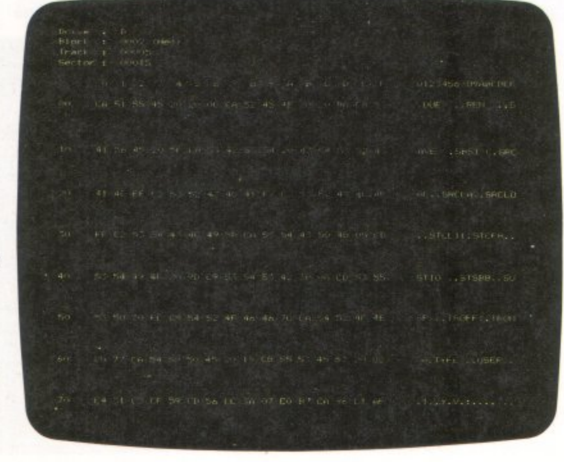
Para ejecutar sus programas en BASIC, en lugar de emplear un interpretador puede utilizar un compilador. Mientras que el interpretador ha de traducir una instrucción cada vez que se le proporcione, el compilador traduce todo su programa, completo, al código de lenguaje máquina del ordenador una sola vez y para siempre. Los programas que han sido "compilados" se ejecutan mucho más rápidamente que el software que ha sido "interpretado".

El BASIC es ideal para escribir programas. Ofrece la ventaja de que es muy similar al lenguaje corriente y muy apropiado para ser utilizado por los principiantes. Pero el programador más experimentado podrá ejecutar sus programas con mayor rapidez utilizando un lenguaje ensamblador. Éste no se parece en absoluto al lenguaje común y, por lo tanto, el programador ha de poseer un conocimiento bastante detallado de cómo realiza sus funciones el microprocesador.

Todas las instrucciones que se le dan al ordenador poseen un equivalente directo en código de lenguaje máquina. Un lenguaje ensamblador está constituido por una serie de instrucciones abreviadas, como MVI *(Move Immediate)* o JZ *(Jump on Zero)*. Éstas se usan para ayudar al programador a recordar sus funciones.

Si el usuario domina un lenguaje ensamblador, lo único que le quedaría por aprender sería el código de lenguaje máquina, pero ello no tendría mucho sentido, a menos que verdaderamente necesitara ganar mínimas fracciones de segundo en el tiempo de ejecución de un programa.

Los códigos de lenguaje máquina de los microordenadores generalmente se escriben de una forma denominada *hexadecimal*. Se trata de una forma de numeración de base 16. Se cuenta de 0 a 9 normalmente, y



El sistema operativo de disco
Cuando los programas se almacenan en un disco flexible, la información se distribuye al azar alrededor de la superficie del disco. El sistema operativo de disco es un programa que automáticamente lleva el registro de la localización de cada byte de información. La ilustración muestra la información almacenada en una sección pequeña de un disco. En el bloque izquierdo, ésta se representa en forma hexadecimal y, en el bloque de la derecha, figura el carácter equivalente. Los códigos que no corresponden a caracteres que han de imprimirse aparecen en forma de puntos

luego se continúa utilizando las letras del alfabeto desde la A a la F para los números del 10 al 15. Cada dirección de memoria comprende ocho dígitos binarios (bits) y éstos se pueden representar mediante un par de dígitos hexadecimales.

Por ejemplo, el número binario 01011101 se ha de dividir primero en dos mitades: 0101 y 1101. Éstas se traducirían en los números decimales 5 y 13, que en el sistema hexadecimal equivalen a 5 y D. Así, al programar en código de lenguaje máquina, 01011101 es 5D. Éste constituye el procedimiento más lento de desarrollar programas, pero probablemente sea la manera de obtener los tiempos de ejecución más rápidos. Lo que sigue corresponde a extractos tomados de programas típicos:

```
BASIC
100 INPUT "Entre las horas trabajadas"; HORAS
200 PAGA = HORAS * TASA
```

La primera línea visualiza en pantalla un mensaje que invita al usuario a dar entrada a las horas trabajadas. Acepta la entrada y luego, en la segunda línea, la multiplica por la tasa de la paga (introducida previamente) para dar la cifra global de la paga.

```
Lenguaje ensamblador
MVL C, 01
CALL 05
```

La primera de las instrucciones anteriores desplaza el valor "1" a una parte de la memoria del microprocesador denominada "registro C". La segunda instrucción entrega el control al sistema operativo. Entonces el sistema operativo vuelve a entregar el control a su programa. Lo que sigue es la traducción directa de las instrucciones en lenguaje ensamblador que citamos arriba:

```
Código de lenguaje máquina
0E01
CD0500
```

Una traducción en código de lenguaje máquina de las dos sentencias en BASIC comprendería muchas órdenes. Evidentemente, es preferible que este trabajo lo realice un compilador o un interpretador, en lugar de escribirlo en lenguaje ensamblador o en código de lenguaje máquina.

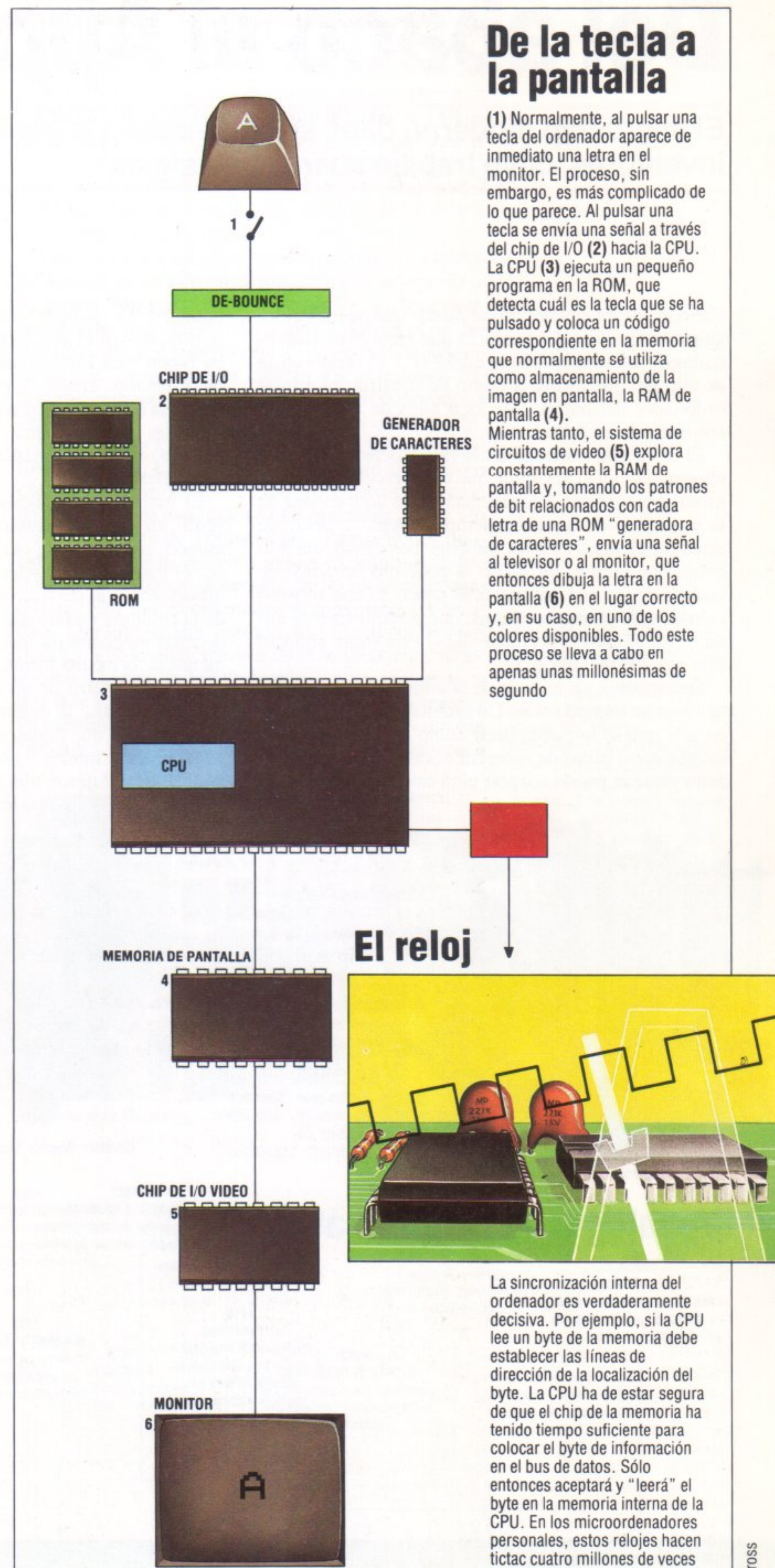
Algunas órdenes del sistema operativo son tan enigmáticas como los programas en lenguaje ensamblador. He aquí algunos ejemplos tomados del CP/M (*Control Program for Microcomputers;* programa de control para microordenadores):

```
DIR * . BAS
```

Esto significa "liste todos los ficheros de la unidad de disco cuyo sufijo sea BAS". DIR es la abreviatura de "directorio".

A pesar de esta forma más bien extraña de elaborar las órdenes, el CP/M está instalado en gran número de máquinas. Ofrece enormes ventajas para los escritores de software profesionales. Los programas se pueden transportar fácilmente de una máquina a otra, con la única condición de que estén escritos de manera tal que entreguen al CP/M el control de la manipulación de los discos, el teclado, la impresora y la pantalla.

Los fabricantes que presenten una máquina nueva pueden beneficiarse del software de base ya existente, y los interesados pueden adquirir su ordenador en la confianza de que con toda seguridad podrán satisfacer sus necesidades en cuanto a software.



De la tecla a la pantalla

(1) Normalmente, al pulsar una tecla del ordenador aparece de inmediato una letra en el monitor. El proceso, sin embargo, es más complicado de lo que parece. Al pulsar una tecla se envía una señal a través del chip de I/O **(2)** hacia la CPU. La CPU **(3)** ejecuta un pequeño programa en la ROM, que detecta cuál es la tecla que se ha pulsado y coloca un código correspondiente en la memoria que normalmente se utiliza como almacenamiento de la imagen en pantalla, la RAM de pantalla **(4)**.
Mientras tanto, el sistema de circuitos de video **(5)** explora constantemente la RAM de pantalla y, tomando los patrones de bit relacionados con cada letra de una ROM "generadora de caracteres", envía una señal al televisor o al monitor, que entonces dibuja la letra en la pantalla **(6)** en el lugar correcto y, en su caso, en uno de los colores disponibles. Todo este proceso se lleva a cabo en apenas unas millonésimas de segundo

El reloj

La sincronización interna del ordenador es verdaderamente decisiva. Por ejemplo, si la CPU lee un byte de la memoria debe establecer las líneas de dirección de la localización del byte. La CPU ha de estar segura de que el chip de la memoria ha tenido tiempo suficiente para colocar el byte de información en el bus de datos. Sólo entonces aceptará y "leerá" el byte en la memoria interna de la CPU. En los microordenadores personales, estos relojes hacen tictac cuatro millones de veces por segundo

Steve Cross

Del ábaco al Apple

El microchip moderno debe su existencia a la genialidad de los inventores, cuyo trabajo abarca tres siglos

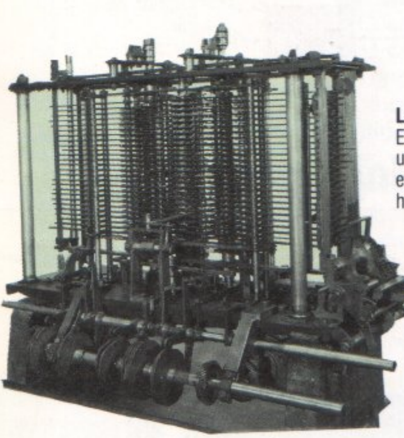
Cuando se está enamorado se piensa que puede conseguirse todo; al menos eso es lo que creía el matemático inglés Charles Babbage en 1830. La consecuencia de ello fue que estuvo a punto de construir el primer ordenador del mundo con cien años de adelanto en relación con su época.

El diseño de Babbage tenía varios defectos: uno de ellos residía en el hecho de que el ordenador debía ser mecánico; otro era consecuencia de las limitaciones de la ingeniería de aquellos tiempos. Pero, a pesar de estos problemas, Charles Babbage construyó una máquina que impresionó tanto al gobierno británico de aquella época que enseguida le concedió una subvención de 1 500 libras esterlinas, que posteriormente aumentó a 17 000. (En la actualidad esta suma equivaldría a 1 700 000 libras.)

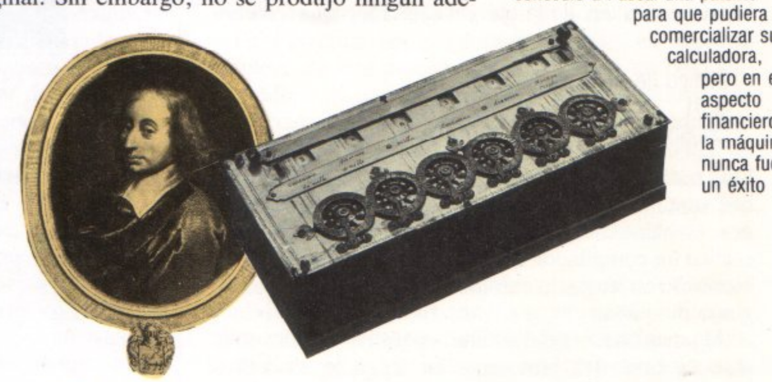
Sin embargo, la historia de la informática se remonta a mucho tiempo antes. Un ordenador es una máquina a la que se le puede decir cómo calcular una serie de números, capaz de recordar el cálculo que ha realizado y que se puede adaptar para calcular otra serie de números. El ejemplo más remoto es el ábaco, que ya se utilizaba en el año 2000 a. C. y que todavía se usa en Japón y en Europa oriental. El ábaco es una clase de ordenador muy útil, porque uno puede ver físicamente la suma en los alambres: la posición de las bolas movibles constituye una "memoria" de la suma. Pero el ábaco no es automático y no es útil para manejar cifras elevadas.

El matemático y filósofo francés Blaise Pascal inventó en 1642 la primera calculadora mecánica del mundo, posiblemente para complacer a su padre, que era el inspector de hacienda de la zona. La máquina funcionaba a la perfección: transportaba los números de la columna de las unidades hasta la columna de las decenas mediante un mecanismo de trinquete, más o menos de la misma forma en que transporta los números el velocímetro de un automóvil, y era totalmente funcional. Blaise la denominó *pascalina*.

A pesar de que la pascalina no se vendió bien, despertó un gran interés a nivel científico y durante unos años se introdujeron muchas mejoras en la calculadora original. Sin embargo, no se produjo ningún ade-



La máquina analítica
En 1834 Charles Babbage diseñó una máquina analítica que podía efectuar cálculos con números de hasta 80 dígitos. Incorporaba muchas de las características del ordenador moderno. Los "programas" se controlaban mediante tarjetas perforadas y los resultados se imprimían automáticamente. También tenía un "molino" aritmético y dispositivos de almacenamiento separados



Blaise Pascal
La "pascalina" fue la primera calculadora mecánica del mundo y fue diseñada en 1642 por el francés Blaise Pascal. Su deseo original era crear una máquina de dividir y multiplicar, además de sumar y restar. La pascalina se valía de un punzón para hacer mover las ruedas y tenía un mecanismo especial que llevaba los dígitos de una columna a la otra. El rey de Francia le concedió a Pascal una patente para que pudiera comercializar su calculadora, pero en el aspecto financiero la máquina nunca fue un éxito

Cortesía Science Museum

Cronología de la informática

1000 A. C.
EL ÁBACO
Aún en uso hoy en día, calcula y almacena números mediante bolas movibles

1614
LOGARITMOS
John Napier los inventa como medio para simplificar la división en resta y la multiplicación en suma

1623
NÚMEROS BINARIOS
Francis Bacon utiliza por primera vez la aritmética con base 2

1642
MÁQUINA DE SUMAR
Blaise Pascal inventa la "pascalina", primera calculadora mecánica

1671
CALCULADORA
Gottfried W. Leibniz logra construir una máquina que multiplica y divide

1802
TARJETAS PERFORADAS
Joseph Jacquard construye un telar que almacena los patrones de tejido en tarjetas perforadas

1822
MÁQUINA DIFERENCIAL
Charles Babbage produce su primera calculadora matemática

1000 A.C. // 1614 1623 1642 1671 1802 1822

lanto trascendente hasta que Charles Babbage y Ada Lovelace comenzaron a pensar en el asunto.

Charles Babbage nació en 1791 en el seno de una familia adinerada. Dispuso de toda clase de facilidades para acceder a una completa educación, y muy pronto demostró ser un genio de las matemáticas; quizá por la frustración que le producía tener que corregir los muchos errores que hallaba en las tablas logarítmicas, se concentró en la construcción de una máquina que pudiera hacer menos monótona la faena de calcular.

En 1822 presentó en la Royal Astronomical Society su primer modelo de un ingenio "diferencial", una máquina que efectuaba los cálculos necesarios para construir tablas logarítmicas. El nombre proviene de una técnica matemática abstracta conocida como método de las diferencias. La Royal Astronomical Society lo alentó para que creara máquinas nuevas y más perfeccionadas.

Trabajando en equipo con Ada Lovelace, hija de Lord Byron, se embarcó en un proyecto aún más ambicioso: construir un ingenio "analítico". Esta máquina se diseñó para calcular los valores de funciones matemáticas mucho más complejas que las logarítmicas.

La creación de esta máquina estuvo plagada de problemas desde el principio. Simplemente, no funcionaba. Al observar los bocetos del ingenio que se han conservado, se deduce que era enorme y que ocupaba todo el inmenso taller que Babbage había construido en su domicilio. Era preciso tornear de forma especial centenares de rodillos, varillas y engranajes, y la tecnología metalúrgica de aquella época no había alcanzado todavía los niveles necesarios de desarrollo para realizarlo satisfactoriamente. Cuando construyó su maqueta a escala, Babbage pudo soslayar las pequeñas inexactitudes de la máquina, pero, una vez construida, estos mínimos desajustes demostraron ser importantes.

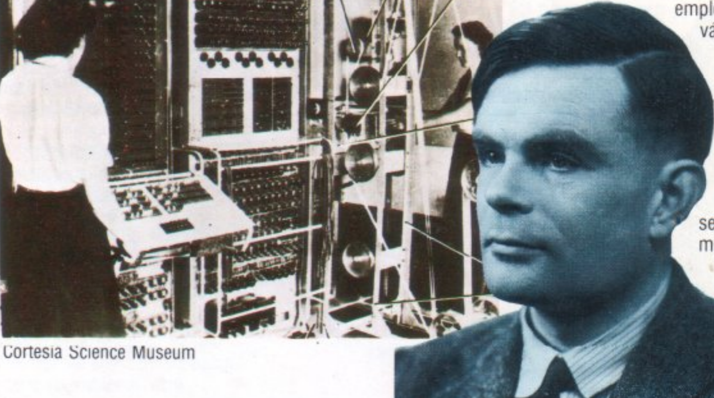
No obstante, Babbage se hallaba en el buen camino y, si hubiera conseguido que todos los componentes del ingenio armonizaran bien, probablemente su máquina analítica hubiera funcionado. Gran parte de la arquitectura lógica y la estructura de diseño de los ordenadores actuales se remonta al trabajo de Charles Babbage, a quien se considera como uno de los padres de la informática moderna.

Uno de los presentimientos que alentaron a Babbage durante su trabajo fue la idea de que a su ingenio se lo podría "programar" o "enseñar" a efectuar cualquier tarea matemática. De haber sido capaz de demostrar esto, o si hubiera podido construir una máquina que efectivamente lo hiciera, los victorianos habrían gobernado su imperio mediante un ordenador a vapor.

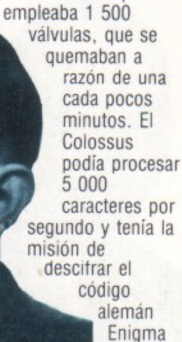
No fue hasta 1936 que se confirmó el presentimiento de Babbage. La prueba de su viabilidad apareció en *On Computable Numbers*, modesta publicación editada por un joven matemático de Cambridge, Alan Turing. El nombre de Turing es prácticamente desconocido para el gran público, pero su contribución fue fundamental para el desarrollo de las ideas cuya concreción hizo que el ordenador se convirtiera en realidad. Durante mucho tiempo los científicos habían pensado que las matemáticas no eran un arte misterioso sino una ciencia exacta regida por reglas lógicas, y que si se le proporcionaba a una máquina esas reglas y un problema, ésta debería ser capaz de resolverlo. Sin embargo, los esfuerzos de los matemáticos más capaci-

El primer programador fue una mujer
La condesa Ada Lovelace, compañera de Charles Babbage y única hija legítima de Lord Byron, es una de las pocas mujeres que han dejado su huella en la historia de la informática. Matemática de enormes aptitudes, comprendió el ingenio analítico ideado por Babbage y escribió algunas de las mejores reseñas acerca de su funcionamiento. Llegó incluso a desarrollar programas para la máquina, convirtiéndose así en la primera persona que programó ordenadores

Cortesía Colección HMG



Cortesía Science Museum



Alan Turing y el Colossus
Alan Turing demostró que una serie de instrucciones sencillas podían resolver cualquier problema complejo. Él y su equipo desarrollaron el Colossus, uno de los primeros ordenadores, que en la fotografía se ve en pleno funcionamiento durante la segunda guerra mundial. Esta enorme máquina empleaba 1 500 válvulas, que se quemaban a razón de una cada pocos minutos. El Colossus podía procesar 5 000 caracteres por segundo y tenía la misión de descifrar el código alemán Enigma

1834
MAQUINA ANALÍTICA
Babbage formula muchas de las ideas sobre las que se basa el ordenador moderno

1835
PRIMER PROGRAMA
Ada Lovelace escribe el primer programa

1847
EL ÁLGEBRA BOOLEANA
George Boole inventa una forma de matemáticas que favorece el estudio del cálculo

1890
PROCESAMIENTO DE LA INFORMACIÓN
Hermann Hollerith utiliza tarjetas perforadas para llevar a cabo el censo de Estados Unidos

1900
LA MEMORIA MAGNÉTICA
Valdemar Poulsen realiza las primeras grabaciones magnéticas de información

1906
LA VÁLVULA
Lee de Forest inventa el interruptor electrónico

1931
EL ANALIZADOR DIFERENCIAL
Vannevar Bush inventa el analizador diferencial, para resolver una familia de ecuaciones diferenciales

1936
LA MÁQUINA DE TURING
El matemático inglés Alan Turing establece los principios teóricos del ordenador

1938
CIRCUITOS ELECTRÓNICOS
Claude Shannon demuestra que los circuitos electrónicos de conmutación pueden realizar operaciones lógicas

1941
CONRAD ZUSE
Construye el primer ordenador electrónico programable

1834 1835 1847 1890 1900 1906 1931 1936 1938 1941

tados por desarrollar una máquina de esas características habían resultado vanos. Turing decidió enfocar el problema desde una perspectiva diferente. Estudió la clase de problemas que podría resolver una máquina siguiendo reglas lógicas e intentó hacer una lista de todas ellas. Si éstas comprendían la totalidad de las matemáticas, la presunción podría confirmarse.

Turing dirigió un equipo de investigación en Buckinghamshire y desarrolló el invento más secreto de la segunda guerra mundial, el Colossus, primer ordenador electromecánico del mundo. Fue esta máquina la que descifró los mensajes alemanes en código Enigma durante el conflicto bélico.

Acabada la guerra, Turing se estableció en Estados Unidos para poner su genio al servicio del primer proyecto norteamericano de informática. Con su ayuda se construyó el primer ordenador norteamericano. Éste se llamó ENIAC y se desarrolló en la Universidad de Pennsylvania. Empleaba 18 000 válvulas, ¡cada una de las cuales se quemaba en dos minutos!

Una de las razones por las cuales el nombre de Turing es virtualmente desconocido, es que trabajaba para la MI6 y tanto su trabajo como él mismo estaban rodeados por el mayor secreto. El gobierno británico no dio a conocer hasta 1975 detalles de la labor pionera de Turing en el campo de la informática.

La informática se perfeccionaba progresivamente, pero no fue hasta 1947, cuando se inventó el transistor de silicio, que el cálculo rápido se hizo realidad.

Los transistores cumplen la misma función de las válvulas, pero a una velocidad superior, con mayor fiabilidad y sin generar calor. Al igual que las válvulas, son interruptores electrónicos que pueden encenderse y apagarse y que se pueden utilizar para representar los ceros y los unos del código binario. Durante los años cincuenta y primeros sesenta se diseñaron ordenadores cada vez más rápidos y de mayores dimensiones, que eran utilizados tanto por las grandes empresas como por los gobiernos.

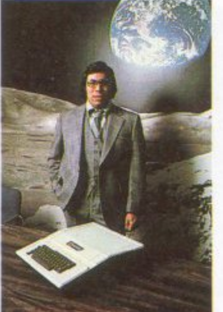
Mediada la década de los sesenta, los científicos comprendieron que un circuito electrónico miniaturizado trabajaría con similar eficacia. Respaldados por los miles de millones de dólares invertidos en la carrera espacial, los laboratorios comenzaron a experimentar colocando diseños de circuitos en un único chip de silicio y grabando luego en él el diseño. Antes de que finalizara la década había nacido este "circuito integrado" y la informática experimentó un impresionante salto adelante. El desarrollo de un "circuito dentro de un chip" llevó, naturalmente, a un "circuito múltiple dentro de un chip", y la inevitable consecuencia de disponer varios chips juntos por capas condujo a la invención del microprocesador.

Aunque la tecnología de los microchips guarda poco parecido con el gigantesco ingenio analítico que construyeron Babbage y Ada Lovelace, y apenas si se parece un poco más al Colossus de Turing, la "arquitectura" práctica que creó Babbage aún se utiliza en el microprocesador actual. La teoría que la hizo posible, la comprobación matemática de Turing de la viabilidad de la informática, aún no ha sido superada.



El "arquitecto" del ordenador moderno
A John von Neumann se le invitó a colaborar en el proyecto ENIAC como asesor en los problemas del diseño lógico. Su informe fue decisivo en el posterior desarrollo del ordenador. Von Neumann aconsejó que los ordenadores futuros almacenaran internamente sus programas y que tanto la información como los programas se representaran mediante números binarios

Cortesía de la London Mathematical Society



Steve Wozniac
En la industria de los ordenadores, a Steve Wozniac con frecuencia se le considera una leyenda viviente. En la fotografía aparece junto a su primer invento, el Apple I (hoy una pieza de coleccionista). Wozniac diseñaba ordenadores en la escuela. Aunque nunca estudió ingeniería, fue él quien se percató de que, para que el ordenador se convirtiera en un aparato personal y de precio asequible, era necesario miniaturizarlo. El Apple II, que desarrolló en 1976, dispuso de la primera unidad de disco para ordenador personal y actualmente sigue siendo uno de los modelos más vendidos en todo el mundo

El Sinclair Spectrum
Desarrollado sólo 40 años después de aquellos incómodos primeros ordenadores que ocupaban habitaciones enteras, el Spectrum es pequeño, compacto y económico, y se constituyó en el primer ordenador personal en color puesto a la venta a un precio módico. Los primeros ordenadores eran máquinas gigantescas y complicadas, construidas por los gobiernos e instituciones del estado, y



funcionaban a puerta cerrada. En la actualidad, gracias a máquinas como el Spectrum, es posible que muchas familias puedan comprar un ordenador personal

1943 COLOSSUS
Primer ordenador electromecánico del mundo

1946 ENIAC
Primer ordenador a válvulas (Estados Unidos)

1947 VON NEUMANN
Su informe sobre el ENIAC ejerce gran influencia en el diseño y desarrollo de los ordenadores

1948 EL TRANSISTOR
Lo inventa William Shockley

1949 MARK I: UNIVERSIDAD DE MANCHESTER
Se construye el primer ordenador que responde al diseño de Von Neumann

1951 ACE
Uno de los primeros ordenadores diseñados para cálculo veloz

1957 FORTRAN
El primer lenguaje para ordenador de alto nivel

1957 IBM
Crea su primer ordenador

1962 DISCOS MAGNÉTICOS
Se usan por primera vez con el ordenador ATLAS

1963 CIRCUITOS INTEGRADOS
Se fabrican desde 1963

1964 BASIC
Se inventa el lenguaje para micros más popular

1972 LSI
Large Scale Integration (circuito de integración a gran escala): llega el chip

1982 SPECTRUM
Primer microordenador popular en color

1990 QUINTA GENERACIÓN
Los ordenadores se programan a sí mismos

1943 1946 1947 1948 1949 1951 1957 1962 1963 1964 1972 1982 1990

BBC Modelo B

Gracias a su excelente configuración técnica, avalada por los recursos de comercialización y programación de la BBC de Londres, este microordenador se ha hecho muy popular

El BBC Modelo B, recientemente introducido en el mercado español, lo fabrica Acorn Computers en Cambridge (Inglaterra) y lo comercializa la BBC. Originalmente había dos modelos, pero el Modelo A, más barato y menos sofisticado, está quedando desfasado.

En Gran Bretaña, el Modelo B se ha vendido muy bien en las escuelas y figura en la lista de ordenadores aptos para esa finalidad aprobada oficialmente por el gobierno. Para esta máquina se ha desarrollado una cantidad importante de software educativo, desde lenguajes de programación hasta paquetes de programas para "aprendizaje mediante ordenador".

Las especificaciones técnicas del Modelo B se consideran excelentes. En particular, el lenguaje de programación BASIC del BBC está muy bien equipado con órdenes para afrontar las funciones especiales. También simplifica la tarea de desarrollar y editar programas.

Existen ocho modalidades diferentes para gráficos. Esto significa que el usuario puede elegir entre una resolución alta, media o baja, aunque en el primer caso el número de colores disponibles es limitado. La máxima resolución que se puede obtener es de 640 × 256. La mayoría de programadores optan por un televisor normal, pero para obtener los mejores resultados en los gráficos del Modelo B es recomendable usar un monitor exclusivo. Existen órdenes para trazar en la pantalla líneas y círculos y crear una gran variedad de imágenes.

Su fuente de alimentación eléctrica va alojada en el interior de la carcasa, lo que le proporciona una apariencia bien proporcionada y autosuficiente; pero las interfaces situadas detrás y debajo de la carcasa son más numerosas que en otras máquinas. Ello significa que existe una gran cantidad de dispositivos idóneos para ampliar el ordenador estándar, incluyendo diversas versiones de unidad de disco, aparte de la que la BBC fabrica especialmente.

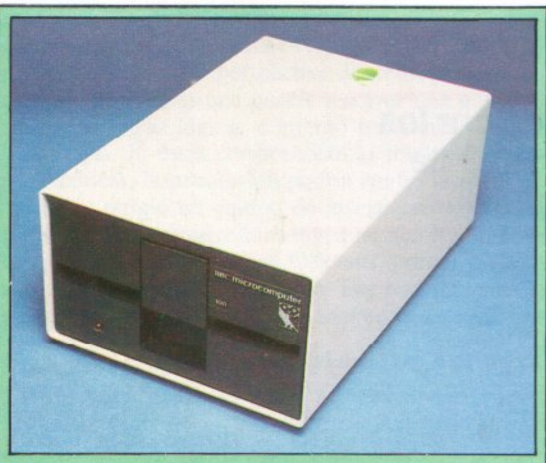
Además de las interfaces para unidad de disco, impresora y un dispositivo análogo, como un equipo de laboratorio o un instrumento de medición, existe una configuración para trabajar en red. Ésta es ideal para trabajar en las escuelas, puesto que varios usuarios pueden compartir la misma impresora o la misma unidad de disco.

Por último, existe "el tubo", una sofisticada interface para conectar un microprocesador alternativo, ya sea para conseguir una mayor velocidad de cálculo o bien para ejecutar el software escrito para otras máquinas. No obstante, al parecer son muy pocos los programadores que aprovechan este dispositivo.



El teclado del BBC Modelo B
El teclado es uno de los puntos fuertes del BBC Modelo B, en términos de trazado, facilidades y calidad de construcción. Las teclas están moldeadas convenientemente, gracias a lo cual hasta una persona que esté acostumbrada a mecanografiar al tacto se sentirá muy cómoda. Las cuatro teclas con flechas sobre el lado derecho son para hacer mover el cursor por la pantalla cuando se editan textos o programas.
La hilera superior, de diez teclas rojas de función programable, es particularmente útil para programas educativos, dado que el usuario debe tan sólo escoger la respuesta adecuada entre 10 posibilidades.
Una característica muy agradable es la inclusión de tres LED (Light Emitting Diodes; diodos emisores de luz) para indicar si el motor de la cassette está funcionando y si se han activado las teclas de retención de mayúsculas





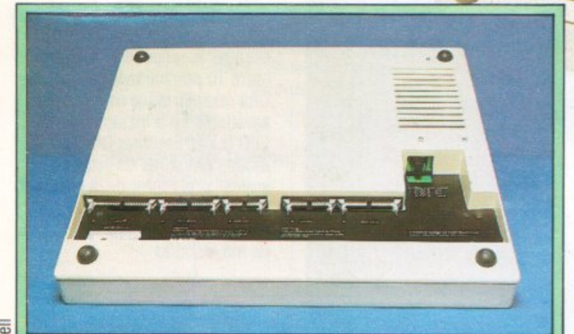
La unidad de disco
La unidad de disco del micro BBC es un accesorio muy atractivo pero caro, y sólo puede almacenar 100 Kbytes. Sin embargo, existe la posibilidad de utilizar las unidades de disco más baratas que proporcionan otros fabricantes

Entrada analógica
Permite que el ordenador lea el voltaje de un componente no digital del equipo, como por ejemplo de un dispositivo sensor de calor. Se utiliza básicamente en los laboratorios y en trabajos de experimentación

Controlador video
Este chip recibe la información de la memoria del usuario y la convierte en una señal de video para que se pueda visualizar

Chips importados
Si observa atentamente los chips, verá que la fabricación de microprocesadores es muy internacional. El BBC Modelo B contiene chips fabricados en Malaysia, Japón, Portugal, Escocia y Estados Unidos

Chips interfaces
Adaptadores versátiles para interfaces, como este MOS 6522, vigilan la conexión por interface con dispositivos externos. Si bien no son procesadores, estos chips son tan sofisticados como el propio microprocesador



Conexiones para periféricos
Vista de la cara inferior del BBC Modelo B, en la que se ven los enchufes en donde se pueden conectar los periféricos

Conexión impresora
Aquí se puede enchufar cualquier impresora de señal en paralelo

Conexión dispositivos
Para la experimentación con dispositivos digitales y circuitos lógicos caseros

Ian McKinnell

Interface cassette
Los programas se pueden guardar de dos formas en una grabadora de cassette corriente: una de ellas proporciona mayor velocidad de grabación y la otra aumenta la fiabilidad de la misma

Salida RGB
Proporciona señales separadas para los componentes rojos, verdes y azules de la señal de video en color. Acciona un monitor en color de gran calidad

Salida televisión
Se acopla con el conector de antena del televisor

Salida video
Para utilizar con un monitor monocromático

Conexión RS232
Interface en serie de gran velocidad para utilizar con determinados periféricos

Modulador
Recibe la señal en color del controlador de video y la convierte en una salida apta para el televisor

Cristal de cuarzo
Un cristal que late y constituye el corazón del reloj, sincronizando todas las operaciones

Microprocesador
El chip MOS 6502 se encarga de todo el procesamiento

RAM
El BBC Modelo B contiene 32 K de RAM para almacenar visualizaciones de programas, datos y gráficos

ULA
Es una matriz lógica, diseñada especialmente, que efectúa el trabajo que en otros ordenadores realizan docenas de componentes. El trozo de metal montado en la parte superior actúa como difusor de calor para evitar el sobrecalentamiento del chip

ROM
Estos dos chips de ROM proporcionan el lenguaje de programación BASIC y el sistema operativo, que es el grupo de programas que se requiere para que el ordenador pueda realizar todas sus funciones internas

Tubo
Una interface especial diseñada por Acorn para que el BBC Modelo B pueda trabajar con microprocesadores alternativos

BBC Modelo B

DIMENSIONES
409 × 358 × 78 mm

PESO
3,70 kg

CPU
6502A

VELOCIDAD DEL RELOJ
2 MHz

MEMORIA
32 Kbytes de RAM, 32 Kbytes de ROM, incluyendo BASIC y un sofisticado sistema operativo

VISUALIZACION EN VIDEO
Ocho modalidades diferentes para gráficos, que proporcionan una gran variedad de visualizaciones. Mayor superficie de textos: 32 líneas de 80 caracteres. Resolución más alta para gráficos: 640 × 256 pixels. Hasta 16 colores con una resolución más baja

INTERFACES
Televisor, monitores monocromáticos y en color, unidad de disco, impresora, entrada analógica, conexión dispositivos, tubo (para conectar microprocesadores adicionales)

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
LISP, FORTH, LOGO

VIENE CON
Cables para cassette y televisor. Guía para el usuario. Folleto y cassette de demostración

TECLADO
Estilo máquina de escribir, con 74 teclas móviles, incluyendo 10 teclas de función programable

DOCUMENTACION
La Guía para el Usuario del microordenador BBC parece haber sido escrita por personas de gran experiencia y que daban por sentado que todos sus lectores ya eran profesionales de la informática.
Varios capítulos se dedican a la utilización especializada de los programas del sistema, que controlan las sofisticadas configuraciones de la máquina para gráficos, sonidos y entrada/salida.
Se incluye una explicación detallada y muy completa de la forma en que opera y está programado el microprocesador 6502 y, a diferencia de muchas de las secciones de este tipo de otros manuales, no se trata tan sólo de una copia de la documentación Rockwell original

Puertas y sumadores

Los números binarios, compuestos de ceros y unos, se pueden sumar entre sí mediante la sencilla lógica del AND, OR y NOT

En un artículo anterior (véase p. 68) hemos visto cómo los relativamente sencillos circuitos de transistor se pueden utilizar para tomar decisiones lógicas como AND, OR y NOT. Lo sorprendente es que estas mismas "puertas lógicas" también constituyen los bloques de construcción que se emplean para realizar las funciones aritméticas en el interior del ordenador. Lógicamente, las entradas de las puertas son, bien de voltaje cero, para representar "falso", o bien de voltaje positivo, para representar "verdadero". La ausencia de voltaje se suele simbolizar con un cero (0) y el voltaje positivo con un uno (1). Cuando las puertas lógicas se utilizan para la aritmética, se emplean los mismos ceros y unos, pero entonces representan literalmente los unos y los ceros que se suman.

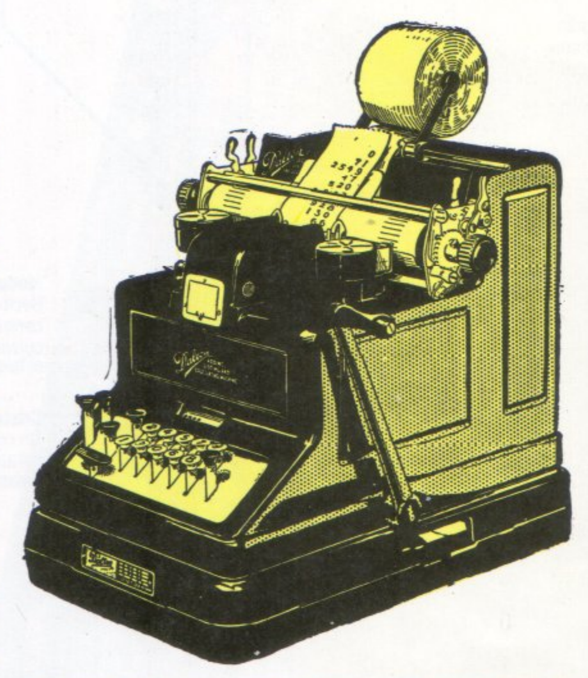
Si deseamos sumar dos dígitos binarios, sólo habrá dos entradas en el circuito de suma, y únicamente podrá haber cuatro combinaciones de entrada: 0 + 0, 0 + 1, 1 + 0 y 1 + 1. Estudiando la aritmética binaria hemos aprendido que 0 más 0 es igual a 0 (como en aritmética decimal). Sabemos también que 0 más 1 (o 1 más 0) es igual a 1 (igual que en aritmética decimal). La diferencia, respecto a la aritmética que hemos aprendido en la escuela, es que en binario 1 más 1 es igual a 0 y llevamos 1. La demostración aritmética de estas cuatro sumas sería la siguiente:

X		Y		Z
0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	10

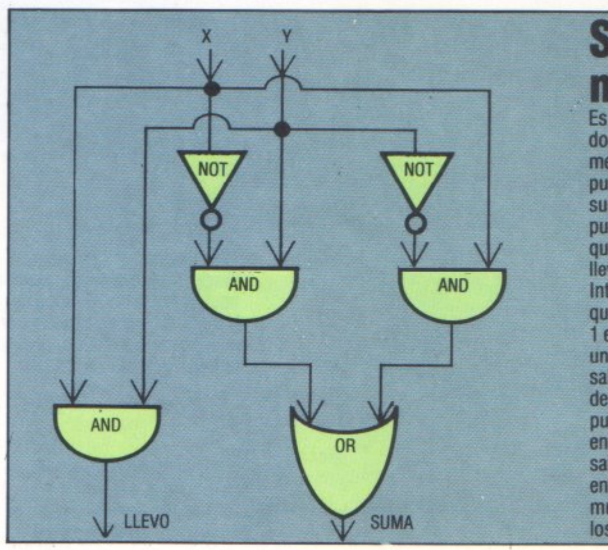
Si hubiéramos de utilizar una puerta OR para efectuar la suma, obtendríamos una salida falsa (0) si ambas entradas fueran falsas (0), y una salida verdadera (1) si alguna de las entradas fuera verdadera (0 + 1 o 1 + 0).

De momento, para sumar dos dígitos binarios parecería perfectamente adecuado utilizar una puerta OR simple. Pero veamos. Si ambas entradas fueran verdaderas, la salida de una puerta OR simple también sería verdadera, pero en aritmética binaria esa respuesta estaría equivocada. La respuesta correcta sería 0 y llevo 1. Una puerta OR simple daría una respuesta correcta para tres de las cuatro combinaciones de entrada posibles, pero tres de cuatro no es un nivel satisfactorio.

Lo que se necesita es un circuito que dé una respuesta de 0 si ambas entradas son 0, y una respuesta de 1 si una de las entradas es 0 y la otra es 1, y una respuesta de 0 si ambas entradas son 1 (como en la tabla de verdad anterior). Esto no es tan difícil como parece. Si tenemos dos puertas AND, con las dos entradas yendo a ambas puertas, pero invirtiendo una de las entradas a través de una puerta NOT por una de las puertas AND, y si la otra entrada se invierte a través de otra puerta NOT que va hacia la otra puerta AND (véase la ilustración), obtendríamos una situación en la cual un 0 en ambas entradas dará una salida falsa en las dos puertas AND, y, del mismo modo, un 1 en ambas entradas dará una salida falsa para ambas puertas. Por otra parte, un 0 en una entrada y un 1 en la otra darán dos entradas verdaderas para una de las puertas AND. Una de estas puertas producirá, por tanto, una salida verdadera. Si las dos puertas AND tuviesen conectadas sus salidas con una puerta OR, la salida de la puerta OR sólo sería verdadera si una, y sólo una, de las dos entradas fuera verdadera.

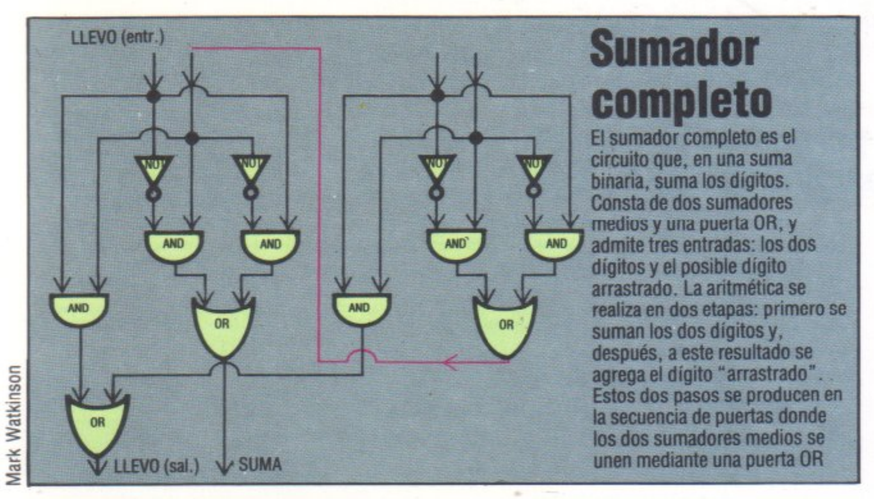


La calculadora de mesa
Hasta que hace poco tiempo se inventara la calculadora electrónica, la máquina de sumar mecánica (o caja registradora) era un objeto muy común en las tiendas y oficinas. Exceptuando unos pocos refinamientos, esta máquina no sufrió modificaciones esenciales durante trescientos años, funcionando mediante una serie de engranajes y ruedecillas. Muy pronto se advirtió el potencial comercial de la calculadora. Pascal inventó la primera máquina de sumar para que su padre, que era inspector de hacienda, la utilizara en su despacho. Cuando Leibniz desarrolló la multiplicación y la división, la calculadora se introdujo en el mundo empresarial



Sumador medio
Es un dispositivo para sumar dos números binarios mediante una combinación de puertas lógicas. Se llama sumador medio porque no puede arrastrar los dígitos que a menudo hay que llevarse al sumar números. Intente sumar 1 y 1. Recuerde que una puerta NOT invierte el 1 en 0 y el 0 en 1. Para que una puerta AND produzca una salida 1, las dos entradas han de ser 1. La salida de una puerta OR será 1 si una entrada o ambas son 1. La salida sólo será 0 si ambas entradas son 0. La ilustración muestra el camino que siguen los dígitos

Mark Watkinson



Sumador completo

El sumador completo es el circuito que, en una suma binaria, suma los dígitos. Consta de dos sumadores medios y una puerta OR, y admite tres entradas: los dos dígitos y el posible dígito arrastrado. La aritmética se realiza en dos etapas: primero se suman los dos dígitos y, después, a este resultado se agrega el dígito "arrastrado". Estos dos pasos se producen en la secuencia de puertas donde los dos sumadores medios se unen mediante una puerta OR

Mark Watkinson

Este circuito está prácticamente allí. El único problema es que una entrada de dos unos, aunque da una "suma" correcta de 0, es incapaz de producir una señal "llevo". Sin embargo, una puerta AND adicional, conectada en paralelo con las dos entradas, producirá una señal "llevo" cuando, y sólo cuando, ambas entradas sean verdaderas. Lo que sigue es la tabla de verdad para el circuito de la ilustración, denominado *sumador medio*:

X (entrada uno)	Y (entrada dos)	LL (salida "llevo")	S (salida "suma")
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Se denomina *sumador medio* porque, en cierto sentido, sólo es adecuado a medias. Es apto para el caso de que todo lo que deseemos hacer sea sumar una única columna de dos dígitos binarios. No obstante, por lo general desearemos sumar dos bytes de datos, y cada byte contiene ocho dígitos. El sumador que se cuidara de la columna de dígitos binarios situada más sobre la derecha, bien podría ser un sumador medio. Sin embargo, todos los demás sumadores a su izquierda habrían de aceptar tres entradas: los dos dígitos de "su" columna y el que se lleve desde la columna situada inmediatamente a su derecha. Observemos esta suma:

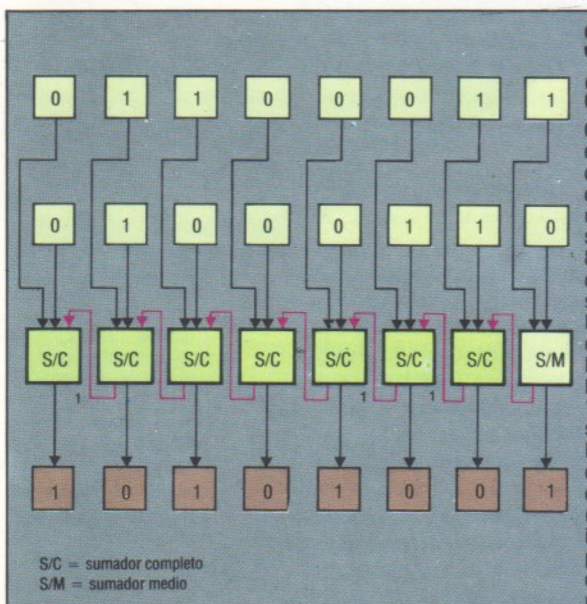
```
 011
+111
1010
```

Al sumar la columna de "los unos" decimos 1 más 1 es 0, llevamos 1 y escribimos un 0 bajo la columna de "los unos". Cuando sumamos la columna de "los dos", decimos 1 más 1 es 0, llevo 1, más lo que nos llevamos de "los unos", que es 1, llevo 1. Escribimos un 1 bajo la columna de "los dos" y nos llevamos 1 a la columna de "los cuatros". Aquí decimos 0 más 1 igual a 1, más lo que me llevaba de la columna de "los dos", que es 0, me llevo 1. Colocamos el 0 en la columna de "los cuatros" y llevamos 1, que escribimos bajo la columna de "los ochos". En otras palabras, la tabla de verdad para un *sumador completo* capaz de manipular los dígitos que se llevan, además de los dos dígitos binarios, sería la siguiente:

X	Y	LL (ent.)	LL (sal.)	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Un sumador completo se puede obtener a partir de la utilización de dos sumadores medios y una puerta OR adicional. La "salida llevo" de cada sumador completo se conecta directamente con la "entrada llevo" del sumador a su izquierda, y de esta forma se pueden encadenar tantos sumadores completos como sea necesario.

En los microordenadores modernos, la mayoría de las sumas y otras operaciones aritméticas se efectúan con grandes cantidades de circuitos sumadores conceptualmente idénticos a los que hemos descrito anteriormente. En su mayoría, estos circuitos sumadores están incluidos, formando sólo una parte de él, en el sistema de circuitos de la CPU. Antes de que llegara la época de la integración a gran escala, que culminó en el microprocesador, comúnmente se utilizaban circuitos integrados más sencillos que apenas contenían unas pocas puertas. Estos circuitos se denominan chips TTL. (TTL significa *Transistor-Transistor Logic*: lógica de transistor a transistor, que alude a la forma en que efectúan la conmutación lógica los transistores que están acoplados entre sí directamente.) El interior de una CPU típica consiste en un único chip de silicio que tiene incorporadas pequeñas zonas de memoria RAM y ROM, grandes cantidades de circuitos de conmutación y una parte denominada ALU o *Arithmetic and Logic Unit* (unidad aritmético lógica). La ALU es la parte de la CPU que contiene todas las puertas lógicas y los sumadores necesarios para que el ordenador efectúe cálculos y tome decisiones lógicas.



Sumador de 8 bits

Cuando se trata de números elevados, usted suma cuidadosamente dígito por dígito. El ordenador hace lo mismo. En la ilustración se suman dos números binarios de 8 dígitos. La hilera superior contiene el primer número, y la segunda hilera el número que se ha de sumar. Trabajando de derecha a izquierda, al sumar los dos primeros dígitos no habremos de agregar ninguno "arrastrado", de modo que será suficiente un sumador medio. Pero para el segundo par de dígitos y para los siguientes han de usarse sumadores completos (el 1 debajo de los sumadores indica "llevo 1"). Haga la suma por el mismo método que el ordenador y verifique su respuesta en la hilera inferior

Grabando

El gran temor del programador es perder la obra de arte que ha creado en la pantalla. Las cassettes ofrecen la solución

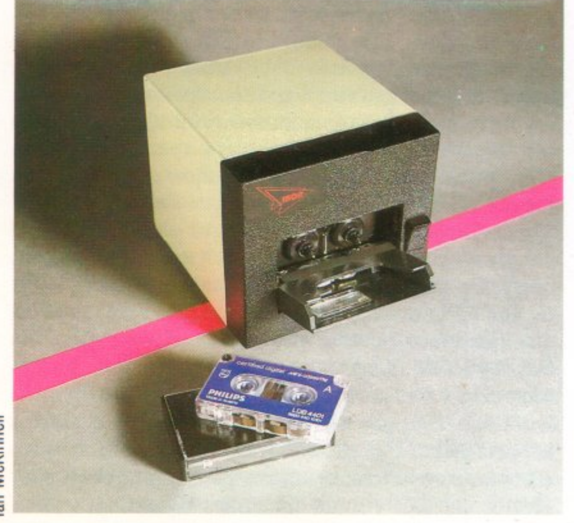
Altavoz
En la mayoría de las grabadoras de cassette el altavoz queda desconectado cuando se acopla el aparato a un ordenador o a un equipo de alta fidelidad

Cabeza de borrado
Estando en modalidad de grabación, elimina de la cinta cualquier señal grabada previamente

Contador de vueltas
Un dispositivo esencial si se piensa almacenar diversos programas en una misma cinta

Cabeza de grabación/reproducción
Esta cabeza de doble función graba las señales sonoras en la cinta magnética y las reproduce

La Hobbit
La Hobbit de la Ikon es una grabadora de cassette exclusiva, es decir, diseñada exclusivamente para almacenar programas de ordenador. Es superior a la grabadora de cassette normal, porque la Hobbit está totalmente controlada por un software. No es necesario pulsar "wind", "rewind", "play" ni "record", puesto que la Hobbit realiza todas estas funciones por sí misma. Si desea cargar (LOAD) un programa, deberá digitar el nombre del mismo y la Hobbit buscará en su propio catálogo hasta localizarlo. Luego rastrea la cinta hasta colocarla en la posición correcta



Ian McKinnell

Eje de arrastre
Eje de gran precisión que rota a una velocidad cuidadosamente controlada para hacer pasar por la cabeza de grabación/reproducción 1 7/8 de pulgada de cinta por segundo

Cuando se digita un programa en un ordenador personal, ya sea tomándolo de un libro, de una revista o confeccionándolo uno mismo, la información se almacena en la RAM. Mientras el ordenador está encendido, el programa está a salvo; pero desaparece en el momento en que usted desconecta la máquina. Esto puede suponerle un gran contratiempo: ¡toda una sesión de programación perdida para siempre! La próxima vez que desee utilizar ese programa habrá de digitarlo todo de nuevo. Para evitar este problema, los fabricantes de ordenadores personales suelen incorporar un procedimiento en virtud del cual se puede almacenar con carácter más permanente el contenido de la memoria del ordenador.

Volumen
Regula el volumen de reproducción y ha de determinarse con sumo cuidado, ya que de lo contrario se corre el riesgo de que las cassettes no se carguen correctamente

Motor
El motor acciona el eje de arrastre a una velocidad constante y también hace girar los tambores para bobinar y rebobinar la cinta

Control remoto
Una configuración muy útil que permite que el ordenador personal controle a la grabadora de cassette

Conexión auricular
Muchos micros emplean esta salida para volver a alimentar los tonos al ordenador

Conexión micrófono
Utilizada a menudo para la entrada de datos del ordenador en la grabadora. No obstante, sólo debe emplearse si no se dispone de conexiones para DIN y auxiliares. Cuando se utiliza esta entrada es necesario realizar un cuidadoso ajuste de los mandos de tono y volumen

David Weeks

El procedimiento más común de almacenamiento es el de la cinta de cassette. Escogido originalmente por su gran disponibilidad y su bajo precio, en la actualidad casi todos los ordenadores personales que se venden incorporan este sistema. La forma en que cada ordenador almacena su información varía ligeramente; por ejemplo, un programa creado y almacenado en un ordenador Commodore no se puede cargar en un Spectrum. Sin embargo, el procedimiento que se utiliza para convertir el programa en una forma almacenable, es casi universal.

La clase de grabadoras de cassette que utilizan la mayoría de los ordenadores personales es, obviamente, más adecuada para almacenar sonidos; sin embargo, en el interior del ordenador el programa se almacena en forma de números binarios. Éstos han de convertirse en sonidos para que el ordenador pueda reconocer la diferencia entre un bit que "se enciende" y un bit que "se apaga": los ceros y los unos binarios. Para hacerlo, el procedimiento más sencillo consiste en crear un sonido que represente un 1 y otro que represente un 0.

Por lo general, estos sonidos se escogen de manera que sean un tono de 2 400 ciclos para el 1 y un tono de 1 200 ciclos para el 0.

Cuando se digita en el ordenador la orden SAVE, lo primero que se grabará en la cinta son algunos segundos de tono constante. Esto se hace para que al reproducirle la cinta al ordenador en algún otro momento éste pueda percibir la diferencia entre la cinta virgen y la sección que aloja el programa. La primera información real que se graba es la serie de tonos que representan a los caracteres del nombre que le hemos dado al programa. Cada carácter está compuesto de un byte (ocho bits en total), de modo que para representar un carácter son necesarios ocho tonos. Sin embargo, para indicar el comienzo y el final de cada byte el ordenador normalmente coloca un tono extra a ambos extremos. Éstos se denominan *bits de principio y final* y su valor siempre es el mismo: ya 1 o 0, según de qué ordenador se trate.

El propio programa se almacena de forma bastante similar, excepto en ocasiones que se lo divide en segmentos. Normalmente estos segmentos son de 256 bytes cada uno y suelen incluir una información adicional que le permite al ordenador asegurarse de que está recargando la información correcta. El sistema empleado aquí es muy sencillo y se denomina *suma de comprobación*. El primer byte contiene el número de bytes de que consta el segmento, y el último byte contiene un número calculado especialmente que representa el total de todos los bytes sumados. Cuando el ordenador lee la cassette verifica si las cifras que figuran en la cinta coinciden con las que él mismo ha calculado y, en caso de que no coincidan, informa del error al usuario.

Algunos sistemas de cassette, como el del modelo BBC Micro, amplían esta comprobación hasta el extremo de otorgar un nombre y un número a cada segmento del programa. En el caso de que se produzca un error, no hay problema alguno, pues se puede rebobinar la cinta unas pulgadas y volver a intentarlo. En notorio contraste, otros sistemas de cassette ni siquiera muestran el nombre del programa que se está cargando.

La velocidad baudio

La velocidad a la cual se producen y se graban los tonos en la cinta se suele denominar, incorrectamente, *velocidad baudio*. El nombre proviene del código Baudot, que se utilizaba en las primeras versiones del telégrafo eléctrico, y que realmente se refiere al número de veces por segundo que cambia la señal. Una medida más exacta sería el número de bits que se graban por segundo. A mayor velocidad (entre 300 y 1 200 bits por segundo), más rápidamente se almacenarán los programas en cinta y menos tiempo se tardará en volver a cargarlos en el ordenador. Lamentablemente, la mayor velocidad de almacenamiento de los tonos incide en la fiabilidad; una velocidad de 1 200 bits por segundo es a la vez fiable y rápida. Algunos sistemas ofrecen dos velocidades, por lo general una velocidad lenta superfiable de 300 bits por segundo y una velocidad rápida de 1 200 o 2 400 bits por segundo. Para evitar imprevistos, los programas valiosos se pueden guardar en dos copias, una de cada tipo.

La cinta de cassette ha de ser de buena calidad; no hay ningún inconveniente en usar las cintas de sonido normales en vez de las cassettes especiales, siempre y cuando sean de una marca prestigiosa y su longitud no exceda de C-60.

La capacidad aproximada de una determinada longitud de cinta se establece dividiendo por 10 la velocidad de la interface para cassette. El resultado de esta división corresponderá a la cantidad de bytes que se pueden almacenar en la cinta por segundo; una C-60 de 30 minutos por cada lado, en la que la interface trabaje a 1 200 bits por segundo, podría retener 432 Kbytes de programa.

Almacenamiento seguro

El ordenador puede almacenar en su memoria miles de bytes de información y recordar dónde está localizado cada uno de ellos

La memoria del ordenador se puede describir en términos de almacenamiento a corto plazo y almacenamiento a largo plazo. La de largo plazo no pierde la información almacenada y la puede retener durante largos períodos, aun después de haberse apagado el ordenador. En esta categoría entran las cintas magnéticas y los discos flexibles.

Los ordenadores también necesitan una memoria rápida a corto plazo para el almacenamiento temporal de programas y resultados.

Otra forma de describir la memoria del ordenador consiste en considerarla en términos de memoria interna o memoria externa. La memoria interna está localizada dentro del ordenador y por lo general es totalmente "electrónica", mientras que la memoria externa es periférica o exterior al ordenador. La memoria externa por lo general es parcialmente mecánica, y comprende dispositivos tales como cassettes, unidades de disco flexible e incluso tarjetas de papel perforadas.

Generalmente la memoria electrónica interna se denomina memoria principal, mientras que la memoria externa se considera como una memoria secundaria o de apoyo. En la actualidad existen dos variedades principales de memoria interna: RAM y ROM.

EPROM
El problema de las memorias ROM normales es que poseen un contenido "incorporado", determinado en una etapa de la fabricación y que no se puede modificar. Las unidades EPROM (Erasable Programmable Read Only Memory: memoria de lectura sólo susceptible de borrar y programar) son mucho más flexibles. Una vez programadas, se pueden volver a programar borrando los contenidos y volviendo a "escribir" en ellas. Las unidades EPROM incorporan una "ventana" de sílice que permite que penetren los rayos ultravioletas, haciendo que se descarguen los condensadores que almacenan los bits en la EPROM. En ausencia de luz ultravioleta, los condensadores conservan su carga indefinidamente y se retienen los contenidos de la memoria

Tanto la RAM como la ROM son dispositivos totalmente electrónicos, fabricados en forma de chips de silicio y empaquetados en estuches plásticos rectangulares con cables paralelos de hojalata o chapados en plata. Existe mucha similitud en cuanto a la forma en que son seleccionados y "dirigidos" por la CPU del ordenador, pero de ello nos ocuparemos más adelante.

La principal diferencia funcional es que los chips de memoria ROM se utilizan para almacenar programas con carácter permanente. El patrón de unos y ceros de cada dirección de memoria se fija y se determina durante el proceso de fabricación y, en consecuencia, no se puede modificar. Las ROM son las "bibliotecas de referencia" del mundo de la informática. El ordenador puede consultar el contenido de la ROM, pero no puede "escribir" nada en ella.

ROM corresponde a las siglas de Read Only Memory (memoria de lectura solamente); la palabra lectura describe lo que hace el ordenador cuando "accede a" o recupera información de la memoria. Existen diversos tipos de ROM que difieren ligeramente entre sí; algunos de ellos permiten eliminar o borrar especialmente el programa interno y se pueden volver a programar. No obstante, una ROM bastante clásica es la 2364 de Intel. Este chip es una ROM de 65 536 bits, organizados en 8 Kbytes de 8 bits. Esto significa que los 64 Kbits están agrupados en bytes de 8 bits y que cada localización "dirigible" accede a o lee un byte entero. En matemáticas, $1 \text{ K} = 2^{10}$ (dos a la décima potencia) o 1 024, de modo que $64 \text{ K} = 64 \times 1\,024$ o 65 536.

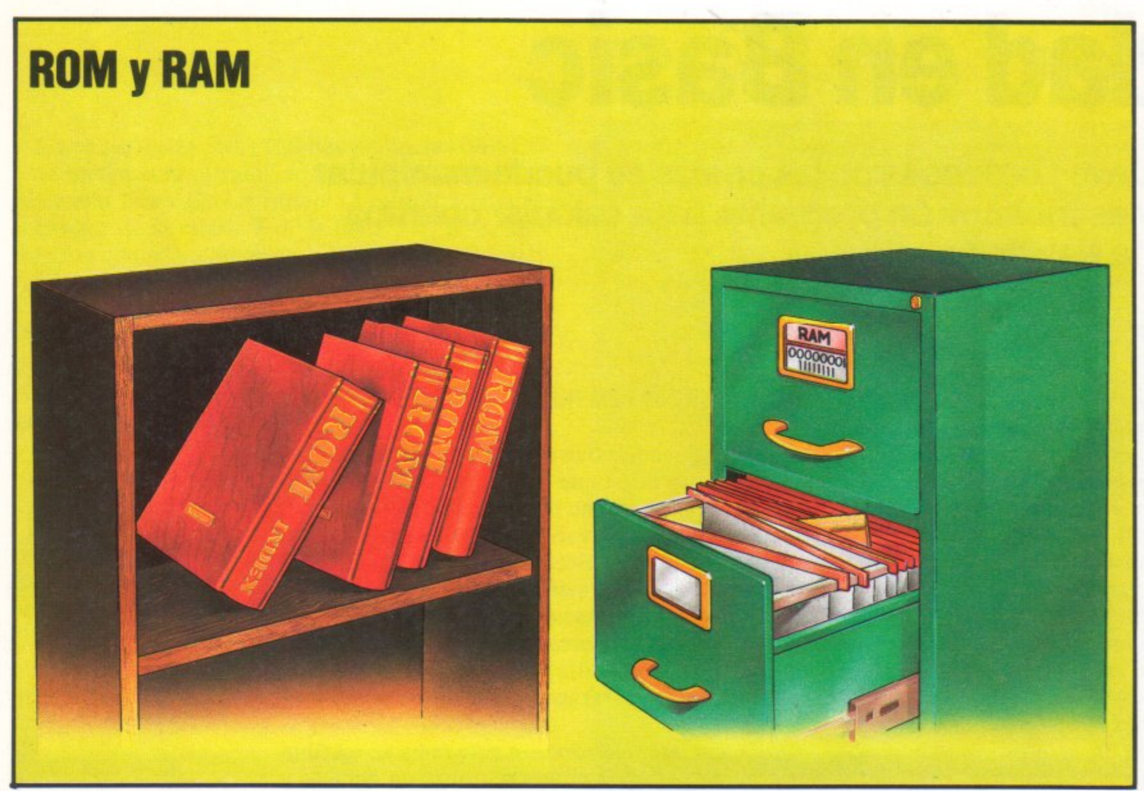
El ordenador, por lo tanto, ha de ser capaz de seleccionar cualquiera de las 8 192 (8 K) localizaciones de dirección. Una atenta lectura de las especificaciones del chip 2364 revela que posee 28 patillas, con una reservada para la fuente de alimentación eléctrica de +5 voltios y otra para la conexión a tierra. Esto deja un total de 26 patillas. Cada byte contiene ocho bits, de modo que cuando se lee un byte del chip, los ocho dígitos de ese byte han de traspasarse mediante cables desde el chip a la CPU. En consecuencia, hay ocho cables para trasladar los bits del byte que se le está leyendo a la CPU. Estos cables se denominan *bus de datos*. Ocho de las patillas del chip se dedican a esta función, una para cada uno de los bits del byte.

Esto deja un total de 18 patillas. Una patilla no es necesaria y no se conecta. Se conserva porque es más fácil fabricar chips con un número par de patillas. Cuatro patillas se utilizan para "seleccionar" el chip de diversas maneras. Éstas son la patilla de "habilitación de salida", la patilla de "habilitación del chip" y las dos patillas de "selección de chip". Estas patillas reciben las señales del ordenador para permitir que el chip sepa cuándo es requerido.

Las 13 patillas restantes son las patillas "de dirección". Cada patilla está conectada a un cable de "bus de direcciones", y éste transporta la dirección del byte requerido, codificada en forma binaria. Trece dígitos binarios pueden dar 2^{13} u 8 192 combinaciones exclusivas de unos y ceros, de modo que las 13 líneas de dirección alcanzan exactamente para seleccionar de forma exclusiva cada uno y la totalidad de los 8 192 bytes almacenados en la ROM.

Las unidades RAM son algo así como las pizarras del mundo de la informática. En ellas los programas y los datos se almacenan con carácter provisional, mientras el ordenador está en funcionamiento, y en ellas también "se escriben" temporalmente los resultados y

ROM y RAM



La memoria a largo plazo
La ROM (Read Only Memory: memoria de lectura solamente) se puede comparar con un libro, dado que se trata de un lugar donde se almacena información con carácter permanente. Tal como no se pueden alterar las palabras de una página impresa, del mismo modo no es posible modificar o suprimir los datos de la ROM

La memoria a corto plazo
La RAM (Random Access Memory: memoria de acceso directo) se parece más a un sistema de archivo que a un libro, puesto que la información se puede modificar y los datos no son permanentes; cuando se apaga el ordenador, la RAM se borra por completo

otros datos. En líneas generales, la memoria RAM es más compleja interiormente que la ROM, porque cada uno de los bits de cada byte de la RAM debe ser capaz de modificarse cada vez que se escribe algo en ellos. Un chip de RAM bastante típico es el Intel 2114. Cada chip de RAM 2114 retiene 4 096 bits de memoria y éstos están organizados en 1 024 *nibbles* (medios bytes) de cuatro bits. Esto significa que cada dirección de localización producirá una salida de 4 bits de datos. Por lo tanto, se necesitarán dos de estos chips para producir un Kbyte completo de información. Cada chip 2114 posee sólo 18 patillas, dos de las cuales se emplean para la toma de tierra y la fuente de alimentación eléctrica. Cuatro se utilizan para las líneas de datos de entrada/salida. Una se usa para la señal de selección del chip (la señal que le indica al chip cuándo se lo requiere o cuándo ha sido "seleccionado") y otra se utiliza para indicarle al chip, una vez que ya ha sido seleccionado, si se está escribiendo en él o si se lo está leyendo. Las diez patillas restantes se emplean para el bus de direcciones. Diez líneas de dirección pueden identificar 2^{10} localizaciones exclusivas, o sea 1 024. Si un ordenador dispusiera de 64 Kbytes de RAM, y si se utilizaran chips Intel 2114, se requeriría un total de 128 chips de RAM, puesto que se necesitarían dos chips para cada Kbyte completo. En la actualidad es más común usar chips de RAM de mayor densidad, que en el mismo espacio alojan más memoria. Utilizando chips de RAM más modernos, como el 4164, se pueden obtener 64 Kbytes de RAM con sólo ocho chips.

Año tras año los chips RAM y ROM van siendo más baratos y más compactos, y en la actualidad es posible que un solo chip contenga 128 Kbits. No obstante, el progreso en cuanto al almacenamiento de densidades aún mayores en un solo chip ya no se produce tan rápidamente como antes. El sistema de circuitos contenido en el diminuto trozo de silicio está llegando a ser tan minúsculo, que las técnicas ópticas que se emplean para "grabar" los circuitos apenas llegan a ser suficientes para ese trabajo. Los chips de memoria de "gran densidad" del futuro probablemente se fabricarán mediante procedimientos de grabación por haces de electrones o por rayos X.

En líneas generales, existen dos tipos de memoria RAM en uso, conocidos como RAM estática y RAM dinámica. Ambos tipos poseen sus ventajas y sus inconvenientes, pero actualmente se utiliza más la RAM dinámica que la estática. Ambas pierden el contenido de la memoria apenas se desconecta la alimentación eléctrica, pero la memoria dinámica requiere que se le "refresquen" los contenidos cada algunos milisegundos. Cada bit de la memoria se ha de refrescar o reescribir sin entorpecer la capacidad de la CPU para acceder a la información que contengan. Esto significa que se debe diseñar un sistema de circuitos de sincronización específico y muy crítico, lo que dificulta la labor del diseñador de circuitos.

La memoria dinámica presenta dos claras ventajas sobre la memoria estática. La dinámica sólo requiere un transistor por bit, frente a los tres transistores que normalmente necesita la memoria estática para cada bit. Esto permite almacenar más memoria en chips más pequeños. La mayoría de los chips de RAM dinámica poseen sólo 16 patillas. La otra ventaja de las unidades RAM dinámicas es que consumen menos energía que las estáticas. Por lo tanto, generan menos calor y requieren fuentes de alimentación eléctrica más pequeñas y más baratas.

La ventaja de la RAM estática radica en la simplicidad del diseño de su circuito. Una vez escritos los contenidos de la memoria, permanecen en ella sin que sea necesario refrescarlos. Cada celda de memoria de un bit requiere tres transistores, de modo que resulta difícil conseguir las elevadas densidades que permite la RAM dinámica. Asimismo, la RAM estática consume más energía, y el calor extra generado dificulta la labor del sistema de refrigeración del ordenador, lo que puede hacer necesaria la utilización de un ventilador, con el consiguiente encarecimiento del diseño.

Navidad en Basic

Introducimos nuevas órdenes con las cuales se puede manipular la información y escribimos un programa para calcular cuántos días faltan para la Navidad

En este programa pasamos revista a todos los temas que hemos incluido hasta ahora en nuestro curso de programación; asimismo, introducimos algunas sentencias de BASIC nuevas y eficaces. La finalidad del programa es calcular el número de días que faltan para que llegue la Navidad.

Si observa el listado del programa, verá que empieza con una lista de las variables utilizadas. Ciertamente, esta práctica no es esencial, pero sí es aconsejable para que sus programas resulten más fáciles de comprender cuando se encuentre con ellos algún tiempo después. Algunas versiones de BASIC permiten dar a las variables nombres largos, por ejemplo DAY (día), en lugar de una única letra, como hemos venido utilizando hasta ahora. Si usted es afortunado y posee un BASIC que admite variables de nombre largo, escoja para ellas palabras que posean algún significado. DAY (día), MONTH (mes) o DAYNUM (día número) son mucho mejor que A, X o D. Si no tiene alternativa posible porque su BASIC no admite variables de nombre largo, conseguirá casi la misma facilidad de lectura listando las variables al inicio del programa.

Al ejecutar el programa, lo primero que aparecerá en la pantalla serán las sentencias PRINT que comienzan a partir de la línea 230. Éstas describen brevemente lo que hará el programa y a continuación preparan al usuario para que digite la fecha en la forma indicada, usando comas para separar día, mes y año.

La primera sentencia desconocida aparecerá en la línea 300. Es una sentencia DIMension (dimensión). Se utiliza para determinar el número de ítems o de elementos de que consta la matriz denominada X. Una matriz, también llamada variable subíndice, es como una variable común; la diferencia estriba en que su caja contiene diversos compartimientos. En la línea 300 estamos creando una variable denominada X que dentro de su caja posee 12 compartimientos. En un próximo apartado retomaremos el tema de las matrices y la sentencia DIM con más detalle.

```
310 INPUT D, M$, Y
```

Esta línea es una sentencia INPUT normal, sólo que espera tres entradas. D es una variable numérica y contendrá la fecha de hoy. Y, otra variable numérica, corresponderá al año. M$ es algo diferente. Es una "variable alfanumérica", característica que se indica mediante el signo $ (dólar). Una variable alfanumérica acepta tanto caracteres del teclado como números. Si, por ejemplo, digitamos 23, ENERO, 1984, a la variable D se le asignará el valor 23, a la variable M$ la serie de caracteres ENERO y a la variable Y el valor 1984.

```
330 GOSUB 560 REM RUTINA 'N.º DEL MES'
```

Esta sentencia indica que el programa debe bifurcarse hasta la subrutina que comienza en la línea 560. Observe también que en la misma línea se ha incluido una REMark (observación). Si en la línea hay espacio, no siempre es necesario colocar las REM en una nueva línea. Esta subrutina en particular la utiliza el programa una sola vez y, en rigor, podría igualmente haberse incorporado en el programa principal. Al convertirla en subrutina no hemos hecho más que separar esta parte del resto del programa.

Originalmente, cuando este programa se escribió por primera vez, para el mes se utilizó un número y esta parte del programa no era necesaria. Luego se decidió permitir que se diera entrada al mes digitándolo como una palabra completa. Para convertir el mes escrito en palabras a su equivalente en número, se escribió por separado el programa extra que ahora constituye esta subrutina. El único cambio que hubo que introducir en el programa principal (original) fue la adición de una sentencia GOSUB. Esta subrutina ilustra con qué sencillez se pueden estructurar los programas en bloques y enlazar entre sí utilizando las sentencias GOSUB y RETURN.

La subrutina en sí misma es muy sencilla, pero ilustra lo inteligente que es el BASIC para manipular series de caracteres. Supongamos que hemos dado entrada a ENERO como la parte mes de la sentencia INPUT. A la variable M$ se le asignaría, entonces, la serie de caracteres ENERO. La primera línea de la subrutina es:

```
560 IF M$ = "ENERO" THEN LET M = 1
```

Esta sentencia compara el contenido de M$ con los caracteres comprendidos entre las comillas dobles. Si son los mismos (como en este caso), la línea continúa para establecer el valor de la variable numérica M en 1. No confunda la variable M con la variable M$. Son distintas. Sólo una de ellas puede contener una variable alfanumérica: ¡la que posee el signo $! Después de verificar si M$ es la misma que ENERO, el programa se dirige a la línea siguiente y verifica si el contenido de M$ es el mismo que FEBRERO. Si no lo es, M no se establece en 2. El valor de la variable M sólo se establece cuando la correspondencia es correcta y ese valor es el mismo que el número del mes: 1 para enero, 3 para marzo y así sucesivamente.

Al llegar a la línea 680 el BASIC retorna (RETURNs) al programa principal, a la línea posterior de la sentencia GOSUB. Se trata de la línea 340. Contiene una REM pero ninguna observación. Se incluye simplemente para espaciar más el programa y facilitar su lectura.

Entre las líneas 350 y 370 hay un bucle FOR-NEXT. Éste incrementa el valor de I, comenzando por 1 y siguiendo hasta 12. La variable I se utiliza como el subíndice de la matriz X de la línea 360. Se la debe estudiar con suma atención.

```
360 READ X(I)
```

READ es una nueva sentencia que hasta ahora no habíamos visto nunca. READ se utiliza siempre con su correspondiente sentencia DATA. La sentencia DATA para esta línea está en la línea 510:

```
510 DATA 31, 28, 31, 30, 31, 30, 31, 31, 30,
    31, 30, 25
```

Estos números, a excepción del último, son la cantidad de días que posee cada mes del año. Las dos líneas equivalen a 12 sentencias LET separadas.

```
LET X(1) = 31
LET X(2) = 28
LET X(3) = 31
LET X(4) = 30
LET X(5) = 31
LET X(6) = 30
LET X(7) = 31
LET X(8) = 31
LET X(9) = 30
LET X(10) = 31
LET X(11) = 30
LET X(12) = 25
```

El bucle establecido en la línea 350 hace que I cuente de 1 a 12 para que podamos sustituir X(I), X(1), X(2), X(3), etcétera.

Antes de retomar este programa, consideremos un programa más pequeño y mucho más simple:

```
10 READ A, B, C
20 LET D = A + B + C
30 PRINT D
40 DATA 5, 10, 20
```

Aquí, la sentencia READ de la línea 10 lee el primer ítem de datos (DATA) de la línea 40 y "escribe" su valor en la primera variable. En otras palabras, asigna valor 5 a la variable A. Luego READ lee el siguiente ítem de datos y lo coloca en la variable siguiente. Este programa hace que A = 5, B = 10 y C = 20. Luego suma estos valores y le asigna ese resultado a la variable D. Después este resultado, 35, se imprime (PRINT) en la línea 30.

Volvamos al programa "navideño". La primera vez que se realiza el bucle que comienza en la línea 350, el valor de I se establece en 1. Por lo tanto, la línea 360 equivale a READ X(1). El correspondiente ítem de datos de la línea 510 es 31 (el primer ítem). En consecuencia, X(1) se establece en 31.

La segunda vez que se realiza el bucle, I se convierte en 2, de manera que la línea 360 equivale a READ X(2). El siguiente dato de la línea DATA es 28. Esto significa que X(2) se establece en 28. De este modo, los 12 "compartimientos" de la variable subíndice X se llenan con el número de días de cada mes, excepto el duodécimo, que sólo tiene 25 días. (¿Se ha dado cuenta por qué?)

```
390 GOSUB 750 REM RUTINA 'AÑO BISIESTO'
```

```
100 REM  LISTA DE VARIABLES
110 REM
120 REM  D = FECHA DE HOY
130 REM  M$ = NOMBRE DEL MES
140 REM  Y = AÑO
150 REM  I = INDICE 1
160 REM  X = MATRIZ DE LOS DIAS DE CADA MES
170 REM  R = DIAS QUE FALTAN
180 REM  M = NUMERO DEL MES
190 REM  L = INDICE 2
200 REM  Z = VALOR ENTERO DE Y/4
210 REM
220 REM
230 PRINT "ESTE PROGRAMA CALCULA"
240 PRINT "CUANTOS DIAS FALTAN"
250 PRINT "PARA NAVIDAD"
260 PRINT
270 PRINT "ENTRE DIAS, MES, AÑO DE HOY"
280 PRINT "P. EJ. 12, JULIO, 1984"
290 PRINT
300 DIM X(12)
310 INPUT D, M$, Y
320 REM
330 GOSUB 560 REM RUTINA 'N. DEL MES'
340 REM
350 FOR I = 1 TO 12
360 READ X(I)
370 NEXT I
380 REM
390 GOSUB 750 REM RUTINA 'AÑO BISIESTO'
400 REM
410 LET R = X(M) - D
420 FOR L = M TO 11
430 LET M = M + 1
440 LET R = R + X(M)
450 NEXT L
460 REM
470 IF R = 1 THEN GOTO 500
480 PRINT "FALTAN";R;"DIAS HASTA NAVIDAD"
490 GOTO 520
500 PRINT "FALTA 1 DIA PARA LA NAVIDAD"
510 DATA 31,28,31,30,31,30,31,31,30,31,30,25
520 END
530 REM
540 REM
550 REM
560 IF M$ = "ENERO" THEN LET M = 1
570 IF M$ = "FEBRERO" THEN LET M = 2
580 IF M$ = "MARZO" THEN LET M = 3
590 IF M$ = "ABRIL" THEN LET M = 4
600 IF M$ = "MAYO " THEN LET M = 5
610 IF M$ = "JUNIO" THEN LET M = 6
620 IF M$ = "JULIO" THEN LET M = 7
630 IF M$ = "AGOSTO" THEN LET M = 8
640 IF M$ = "SEPTIEMBRE" THEN LET M = 9
650 IF M$ = "OCTUBRE" THEN LET M = 10
660 IF M$ = "NOVIEMBRE" THEN LET M = 11
670 IF M$ = "DICIEMBRE" THEN LET M = 12
680 RETURN
690 REM
700 REM
710 REM
720 REM NOTA: ESTA RUTINA NO VERIFICA
730 REM       LOS AÑOS BISIESTOS AL FINAL
740 REM       DE CADA SIGLO
750 LET Y = Y / 4
760 LET Z = INT (Y)
770 IF Y - Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

que verifica si el año al que se ha dado entrada es bisiesto o no. Veamos cómo funciona.

```
750 LET Y = Y/4
760 LET Z = INT(Y)
770 IF Y-Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

Un año bisiesto es aquel divisible exactamente por 4. Si se trata de un siglo, también debe ser divisible por 400 para que se considere como un año bisiesto. En aras de la simplicidad, no hemos intentado verificar el siglo, sino sólo la divisibilidad por 4.

La línea 750 establece el valor de Y en el antiguo valor de Y (del año) dividido por 4. La nueva Y será un número entero si el año es exactamente divisible por 4. De lo contrario, tendrá una fracción decimal.

La línea 760 utiliza la función INT para hallar el valor "entero" de Y. Si el número al que se ha dado entrada para el año era 1985, el nuevo valor de la función INT se redondeará en los números más aproximados al número entero. El número a redondear se coloca entre paréntesis después de INT. Alternativamente, se puede colocar entre los paréntesis el nombre de una variable. De modo que LET Z = INT (496,25) establecería el valor de Z en 496.

En la línea 770 se resta el valor de Z al valor de Y y se verifica si el resultado es 0. De ser así, significa que el año es un año bisiesto (puesto que la nueva Y no posee fracción decimal). En este caso, el programa se bifurca hacia la línea 790 mediante GOTO. La línea 790 le suma 1 al segundo ítem de la matriz (el segundo ítem era 28, el número de días de un mes de febrero normal).

Si el resultado de la resta de la línea 770 no era 0, X(2) se deja tal como está y la subrutina retorna (RETURN) al programa principal, a la línea 400.

La línea 400 es otra REM utilizada para espaciar el programa y facilitar su lectura. La siguiente línea que verdaderamente hace algo es la 410, donde R es la variable que alberga el número de días que faltan. Aquí se establece, en el número de días del mes al que se ha dado entrada, menos el día al que se ha dado entrada. Si, por ejemplo, hemos dado entrada a 12, FEBRERO, 1984, D sería igual a 12 y M sería 2. Por lo tanto, X(M) sería igual a X(2) y el segundo ítem de la matriz X es 29 (se le ha sumado 1 porque 1984 es un año bisiesto). En consecuencia, R se establecerá en 29 − 12, es decir, 17, el número de días que faltan del mes en curso, febrero.

En la línea 420 empieza otro bucle. Éste está diseñado para incrementar el valor de M. ¿Comprende por qué decimos FOR L = 1 TO 11 en lugar de FOR L = 1 TO 12? Si M fuera 2, porque hubiésemos dado entrada al mes de FEBRERO, la línea 430 la incrementará a 3. Luego la línea 440 establece a R, el número de días que faltan, en la antigua R más X(M). Ésta ahora equivale a X(3), ya que M se ha incrementado en 1. El valor de X(3) es 31, el número de días del mes de marzo. La línea 440, por lo tanto, establece el nuevo valor de R en 17 + 31 (17 era el resultado de haber restado 12 a 29). La próxima vez que se efectúe el bucle, M se incrementará a 4 y se le sumará al antiguo valor de R el número de días de abril, X(4). De manera que la variable R se convertirá en 17 + 31 + 30.

El último circuito a través del bucle se produce

11 (en la línea 420). La línea 430 le suma 1 al valor de M por última vez, estableciéndola en 12. El valor de X(12) es el último ítem de la sentencia DATA, 25. Obviamente diciembre no tiene 25 días, pero dado que sólo nos interesa el número de días que faltan para la Navidad, 25 es el número tope del mes.

```
470 IF R = 1 THEN GOTO 500
```

Esta línea simplemente verifica si no falta nada más que un día para Navidad, de modo que obtengamos en pantalla una oración correcta desde el punto de vista gramatical. Si R no es 1 es porque falta más de un día, con lo cual la sentencia PRINT de la línea 480 será gramaticalmente correcta.

Y esto es todo. La versión de BASIC que hemos empleado debería funcionar en la mayoría de los ordenadores (ver el recuadro "Complementos al BASIC"), excepto, quizá, para la subrutina "año bisiesto". El BASIC es muy incoherente en cuanto a la forma de utilizar LET. Si líneas como IF M$ = "SEPTIEMBRE" THEN LET M = 9 no funcionan en su ordenador, puede reescribir la subrutina así:

```
560 IF M$ = "ENERO" THEN GOTO 900
570 IF M$ = "FEBRERO" THEN GOTO 910
580 IF M$ = "MARZO" THEN GOTO 920
.
.
900 LET M = 1
905 RETURN
910 LET M = 2
915 RETURN
920 LET M = 3
925 RETURN
(... y así sucesivamente)
```

Esta solución ocupa más espacio y, con tantos GOTO y RETURN, es más difícil de seguir. Sin embargo, demuestra que por lo general existen diversas maneras de solucionar cualquier problema.

Complementos al BASIC



Si el programa se ha de ejecutar en un ordenador BBC, Dragon u Oric, se debe modificar la línea 300 para que diga: 300 DIM X(13)



En el Spectrum no se puede dar entrada a una serie de valores separados mediante comas. Por tanto, se debe modificar la línea 310 de la siguiente manera: 310 INPUT D, y agregar las líneas 312 INPUT M$ y 314 INPUT Y. Como a la fecha no se le da entrada en tres etapas, se deben modificar las sentencias PRINT para que digan: 280 PRINT "¿DIA?", agregando 311 PRINT "¿MES?" y 313 PRINT "¿AÑO?"



Cuando las sentencias REM se colocan al final de otra sentencia, se deben separar mediante dos puntos (:) si se utiliza un BBC, un TI 99/4A o un Spectrum. Por ejemplo, la línea 330 dirá: 330 GOSUB 560: REM RUTINA 'NUMERO DEL MES'



El Spectrum no dispone de sentencia END, de modo que se ha de modificar la línea 520 para que se lea: 520 GOTO 1000, agregando la línea 1000 REM FIN DEL PROGRAMA